Computational Models for the Forensic Analysis of Human Voice

Submitted in partial fulfillment of the requirements for

the degree of

Doctor of Philosophy

in

Department of Electrical and Computer Engineering

Wenbo Zhao

B.S., Process Equipment and Control Engineering, China University of Petroleum
M.S., Electrical and Computer Engineering, Carnegie Mellon University

Carnegie Mellon University
Pittsburgh, PA

August, 2022

Acknowledgments

This Ph.D. thesis would not have been possible without the help and guidance of many. I dedicate this thesis to all those who have kindly helped and guided me in all possible ways. I want to acknowledge their contribution and indispensable role in this thesis and express my gratitude and thankfulness to them from the bottom of my heart. The list below is given by order of time, not importance.

I thank Dongya Zhao, Professor at the China University of Petroleum. He led me to the academic world and inspired and motivated me to pursue graduate studies. He is not good at expressing himself, but he is always proud of having me on his research team. Sadly, I did not return to his team after my graduate studies. However, I will always keep this option open and hope that someday I can return to the beautiful city of Qingdao and meet him again.

I want to thank Ming Li, Professor at Duke Kunshan University. He was my first Ph.D. advisor and the person who gave me the opportunity to be a Ph.D. student. I was an undergraduate from a mechanical engineering background with little knowledge of electrical engineering and computer science. However, Prof. Li took a bold leap and recruited me as one of his first Ph.D. students. We started from a small lab at SYSU-CMU Joint Institute of Engineering (JIE), co-located in Guangzhou and Shunde (a small but peaceful city close to Guangzhou). Ming was a new (Ph.D.) grad with little experience managing a lab and projects. I struggled a lot, learning everything from scratch and spending countless sleepless nights. We struggled a lot, starting a nondestructive ultrasound pipeline inspection project from scratch with little knowledge or experience. We even went through the absurd process of government bidding, negotiating with suppliers, and procuring materials, sensors, and equipment. I designed the blueprints, worked out the technical specifications for a pump-pipeline system, and worked with contractors to eventually deploy the system in a room at our institute building (which took the place of our gym). Ming's speech lab also grew and expanded. When everything went smoothly, the SYSU-CMU JIE mysteriously

shut down following the arrival of SYSU's new principal. Much effort was spent, yet funding was gone, and our pipeline project was doomed. Our ultrasound lab (SMIIP) collapsed, and Ming (and many other faculties and students) sought opportunities elsewhere. However, it was a once-in-a-lifetime setting, and I learned tremendously from this precious opportunity. I honed my skills and grew more professional and mature. Ming is a marvelous advisor and a reliable friend. He helped me in many ways beyond his scope as an advisor. I could not express my gratitude enough. I want to thank Prof. Jimmy Zhu, my co-advisor at CMU; Prof. José Moura at CMU; Prof. Yuanwei Jin at the University of Maryland; Xue Qiu, Ming's wife. Thank you all for your help, guidance, and care. I would also like to thank my lab mates and friends: Uzair Ahmed, Weicheng Cai, Zhun Chen, Duo Cui, Wei Fang, Gaoyuan He, Wenbo Liu, Tao Lu, Han Mei, Zhidong Ni, Luting Wang, Ruiyang Yan, and Jianwen Zhou. I will forever treasure our precious memories at Shunde, the lake in Shunfengshan park, the gate of SYSU, and the college town.

I want to thank Rita Singh and Bhiksha Raj, Professors and my advisors at CMU. They kindly took me in when I had nowhere to go and gave me a second chance to continue my Ph.D. They are everything nice you would expect from an advisor. They are professionally accomplished, academically talented, and are among the most brilliant people I have met. All I have learned and achieved in this thesis work and beyond are attributed to them, for which I will be forever indebted. What they have taught me and their ways of thinking will continue to influence me throughout my life. My gratitude is beyond words. I am more than grateful for their hard and back-breaking work to support my Ph.D., but they did not utter it. I hope I did not let them down, not being the most productive student. Rita and Bhiksha are also supportive family and friends. They care about us. I loved the dinners and (non-drinking) get-togethers they invited me to. I hope I can return to Pittsburgh and meet them often at their offices in the Gates building or anywhere in the world. Their advisory ends, but they continue to be my family and friends. I want to thank Prof. Richard Stern at CMU and my thesis committee: Prof. Rita Singh (Chair), Prof. Bhiksha Raj, Prof. Joseph Keshet, and Dr. Pedro Moreno. Thank you all for helping and supporting me in materializing this

thesis work. I would also like to thank my lab mates and friends: Benjamin Elizalde, Yang Gao, Mahmoud Al Ismail, Abelino Jimenez, Wenbo Liu, Shahan Ali Memon, Yandong Wen, Yangyang Raymond Xia, and Hira Yasin. Thank you all for making my time at CMU and Pittsburgh the most remarkable and memorable experience.

Moreover, I would like to thank my funding resources. The research in this thesis was funded by a gift from Schmidt Sciences in Palo Alto, California. This work would not be possible without the generosity and philanthropy of Schmit Sciences.

Lastly, I would like to thank my family, my parents, my grandma, my sister, and my girlfriend. No words can express my gratitude. I hope my grandpa in heaven can see the world through my eyes and be proud of me, as he always wanted me to achieve something big. I would also like to express my special thanks to my friends: Jianxiao Ge, Peijie Li, Xi Liu, Zhiqian Qiao, Rongye Shi, Chaoyang Wang, Yan Xu, Boyuan Yang, Weikun Zhen, Chenchen Zhu, Yang Zou, and many others.

I have been through a lot since COVID-19, which almost made me give up my Ph.D. In retrospect, I am grateful for these experiences. Good or bad, they made me grow and understand, and treasure whatever life has to gift. I wish my advisors, teachers, family, friends, and myself all the best.

Abstract

Voice-based forensic profiling of humans refers to deducing a speaker's information or characteristics from their voice samples. Specifically, it refers to the set of methodologies, technologies, and tools that represent and model human voices, as well as infer the physical, physiological, psychological, medical, demographic, sociological, and other bio-parametric traits (*bio-relevant parameters*) of a person from their voice.

Voice-based forensic profiling of humans is done based on collected objective evidence that relates measurements made from the voice signal to various bio-relevant parameters of humans. These relations are gauged using a broad spectrum of interdisciplinary technologies and investigative procedures that give us insights and information about these from different perspectives.

Numerous studies from multiple fields in acoustics, speech processing, signal processing, medicine, and psychology have revealed that the human voice carries an enormous number of bio-markers that are unique to the speaker and correlated to the speaker's bio-relevant parameters. Such parameters include physical traits such as age, height, weight, facial skeletal contour, physiological traits such as heart rate, blood pressure, illness, psychological traits such as emotions, mental diseases, and deviation from normal mental states, to name a few. These traits are inherent in the physical articulatory instrument and phonation process and the cognitive and mental processes that influence voice production. As a result, the evidence derived from voice can be distinctive and accurately represent bio-relevant parameters. Profiling attempts to deduce these in a manner that is language/context-agnostic and robust to disguise or fabrication.

In order to deduce bio-relevant parameters from voice, one must develop the appropriate set of voice processing and modeling methodologies. With recent advances in speech processing technologies, many methods and tools have emerged that can potentially be successfully used in this context. For instance, signal processing techniques are used to process raw speech, represent speech, and derive acoustic features

from speech; machine learning and deep learning models are used to model speech and predict the speaker's identity, age, and emotion; dynamical systems are used to model voice production and characterize changes and abnormalities in voice; to name a few.

This thesis aims to develop computational models of voice characterization that are more powerful, more efficient, and more effective in extracting and representing useful information from the voice for forensic profiling. In this thesis, we investigate three categories of models: (1) target-specific models, (2) data-specific models, and (3) process-specific models. *Target-specific models* are tied to a specific task, e.g., predicting a speaker's identity, age, or height from their voice. In this category, we develop supervised machine learning and deep learning models to represent and model human voices such that the target can be best predicted. *Data-specific models* are not bound to a specific task but aim to extract generic information from the voice that can be applied to multiple profiling tasks. In this category, we develop generative models to distill intrinsic data representations, called the "latent features," from the voice signal. We also explore how the algebraic and geometric structure of the corresponding latent feature manifolds aid in target-specific tasks. *Process-specific models* attempt to represent and model the process of voice production through physical (bio-mechanical) means. In this category, we develop dynamical systems of differential equations that explain or emulate the biomechanics of voice production. This approach examines the associated dynamical systems' phase space behaviors and bifurcation maps to characterize many physiological aspects of the human voice. We aim to develop theoretical formulations and practical algorithms for these three models and validate them with simulations or experiments.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This thesis begins with an introduction to voice-based forensic analysis of humans—deriving human characteristics from voices. For such purposes, we introduce three main categories of modeling approaches.

1.1 Voice-Based Forensic Analysis of Humans

Voice-based forensic analysis of humans (VFAH) aims to derive information about a person from their voice. It refers to a set of audio-enabled methodologies, technologies, and tools that can be used to "forensically profile" the speaker of interest in a language-agnostic manner. Thus the focus is not on speech (defined for our purposes as a voice signal modulated to include linguistic content) but on the voice signal itself. More specifically, forensic profiling from voice is the process that derives a person's physical, physiological, psychological, or other bio-parametric or bio-descriptive traits from their voice by computational methods.

Before discussing VFAH, let us first briefly discuss forensic analysis in general. Forensic analysis (FA) comprises a set of procedures that aim to aid the investigation of legal issues by collecting and analyzing objects and data pertinent to the crime and presenting relevant information derived from it to the investigating authorities [1]. The information is usually presented as objective evidence to a court or law enforcement entity to identify the perpetrators, reasons, causes, and consequences of a breach of

law or a violation of rules. This evidence, digital or on paper, is obtained through a broad spectrum of cross-domain, interdisciplinary technologies and investigative procedures and methods [2]. The source of information being investigated can have enormous breadth, ranging from biological traces such as fingerprints, DNA, other biological prints and matter, to paper traces in documents such as handwriting and artworks, to digital traces such as audio, video, images, and text, all of which may appear on a wide range of platforms such as computers, phones, mobile devices, etc. and a wide range of fora such as websites, newsgroups, social networks, etc [3].

With this understanding, forensic analysis of humans (FAH) is the detailed investigation of humans based on evidence collected from varied sources, intending to reveal their reasons, motivations, and methods of perpetrating crimes, as well as the characteristics that can help identify them. In this thesis, we focus on the latter, wherein forensic analysis may be conducted to describe a person of interest in as much detail as possible—e.g., deducing the person's physical characteristics such as age, height, weight, skeletal ratios, diagnosing the person's physiological conditions such as heart rate, blood pressure, state of health, revealing the person's psychological states such as emotion, mental diseases, behavior, state of intoxication, etc. VFAH is FAH that uses voice-centric methodologies and technologies and derives information from voice evidence.

The use of the human voice as forensic evidence in this context is justified by many studies in the literature carried out in multiple scientific disciplines. These studies have revealed that the human voice carries an enormous amount of information about the speaker at the time of speaking. It carries unique signatures that are correlated to the speaker's physical [4, 5, 6, 7, 8, 9], physiological [10, 11, 12, 13] and psychological traits [14, 15, 16, 17], among many others that we do not explicitly enumerate here. The traits we study are evident in the voice signal produced by the human vocal system and hence are language or context-agnostic [18, 19]. These traits are embedded in the voice signal through the physical workings of a person's articulatory instrument. They are consciously or subconsciously influenced by the person's cognitive and mental states. Furthermore, such evidence derived from voice can be distinctive and yield

accurate information about the speaker's bio-parameters if it is derived in a manner that makes it robust to voice disguise or fabrication.

1.2 Modeling for VFAH

One of the ways in which computational forensics comes to the aid of traditional FA and VFAH is by giving us the ability to process and derive information from large volumes of data efficiently. It also provides other mechanisms that do not rely so much on data but are more scalable and reliable. The main approaches to VFAH from this perspective are based on computational models and algorithms that build on concepts that have been developed in fields, such as signal and image processing, speech processing, computer vision, artificial intelligence, machine learning, deep learning [20, 21, 22], etc.

With the advances in speech and audio processing technologies, many computational methods and tools have become promising in the context of computational VFAH. For instance, signal processing techniques are used to process raw speech, represent speech, and derive acoustic features from speech [23]; machine learning and deep learning models are used to model speech and predict the speaker's identity, age, emotion, or even the facial features [24, 25, 26]; dynamical systems are used to model voice production and characterize changes and abnormalities in voice [27, 28]; to name only a few.

Computational models for VFAH can be divided into three broad categories: (1) target-specific models, (2) data-specific models, and (3) process-specific models.

Target-specific models have a specific analysis target of interest, and they aim to derive information from human voices and make judgments about this analysis target. These models are primarily machine learning models that learn patterns from human voice data and predict (or infer) the target. The target of interest can be, for instance, identity, age, height and weight (and hence BMI), skeletal ratio, facial structure, diseases, emotion, etc.

Data-specific models are target-agnostic models, which are machine learning

models trying to learn intrinsic patterns from voice data independently of any task goals. Such models aim to capture data characteristics (e.g., generating distributions) by modeling observed and confounding variables. These are often captured in terms of the dependency dynamics between a sample space and one or more latent spaces. In this context, generative models can generate samples with the same underlying distribution as the observed data. Common generative models include graphical models such as Gaussian mixture models, hidden Markov models, Bayesian nets, restricted Boltzmann machines, deep neural models such as deep belief nets, variational auto-encoders, generative adversarial nets, etc. Notably, generative models may be able to encode intrinsic data information into some latent representation(s), which in turn can be further used in specific tasks.

Process-specific models are physical models that represent a specific physical process mathematically, often via dynamical systems that comprise coupled differential equations with constraints. These models may explain the observed data through ordinary or partial differential equations, often with appropriate assumptions related to the process's continuity or smoothness. Such models characterize the dynamics of the physical process they model in their configuration space or phase space. One can study the properties and behavior of the process, such as stability, bifurcation, and sensitivity in these spaces.

1.3 Objectives of this Thesis

So far, we have outlined the definition, objective, and approaches for voice-based forensic analysis of humans (VFAH). Among the approaches, computational modeling plays a central role. We are particularly interested in three categories of computational models, as shown in Figure 1-1.

This thesis aims to develop computational models of the human voice that can discover, represent, or extract useful information from the voice signal to profile the speaker accurately. Through this, we also hope to understand these modeling approaches' broader strengths and weaknesses and expect this will significantly help

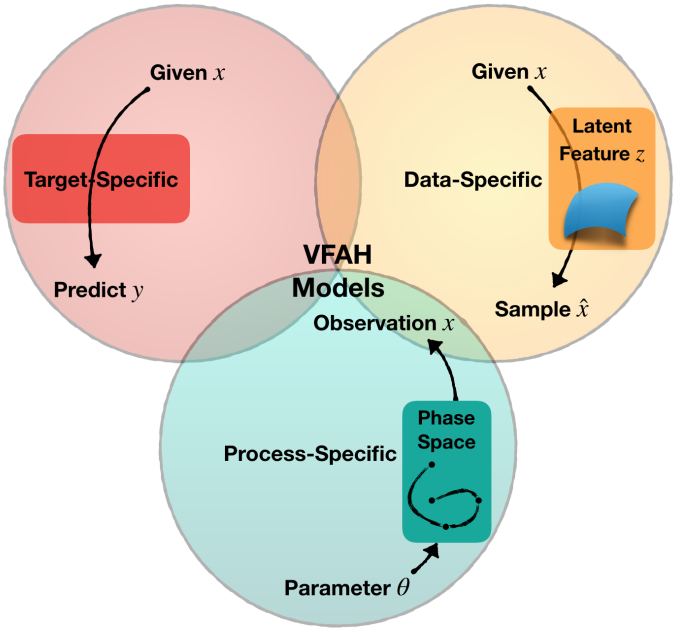


Figure 1-1: Diagram of the three model categories for VFAH.

design optimal approaches for different voice profiling applications. In this thesis, we investigate three types of computational models in depth: (1) target-specific models, (2) data-specific models, and (3) process-specific models. Our specific goals are:

1. In the category of target-specific models, we will develop machine learning and deep learning approaches to model human speech to best predict the speaker's target bio-parameters.

2. In the category of data-specific models, we will develop deep learning-based latent representations for voice signals that will allow us to design or discover features that capture the most intrinsic signatures of human voices in a manner that is most effective for a variety of profiling tasks.

3. In the category of process-specific models, we will develop dynamical system models for voice production and study and utilize the phase space behaviors of these systems to characterize many physiological aspects of the human voice.

We build theoretical formulations and practical algorithms in the three model categories and validate them with relevant, systematically conducted experiments.

References

[1] M. A. Jobling and P. Gill. "Encoded evidence: DNA in forensic analysis". In: *Nature Reviews Genetics* 5.10 (2004), p. 739.

[2] R. Saferstein and A. B. Hall. *Forensic science handbook*. Vol. 1. Prentice Hall Upper Saddle River, 2002.

[3] S. L. Garfinkel. "Digital forensics research: The next 10 years". In: *digital investigation* 7 (2010), S64–S73.

[4] S. E. Linville and H. B. Fisher. "Acoustic characteristics of perceived versus actual vocal age in controlled phonation by adult females". In: *The Journal of the Acoustical Society of America* 78.1 (1985), pp. 40–48.

[5] M. Sedaaghi. "A comparative study of gender and age classification in speech signals". In: *Iranian Journal of Electrical and Electronic Engineering* 5.1 (2009), pp. 1–12.

[6] I. Mporas and T. Ganchev. "Estimation of unknown speaker's height from speech". In: *International Journal of Speech Technology* 12.4 (2009), pp. 149–160.

[7] A. Fedorova et al. "Exploring ANN back-ends for i-vector based speaker age estimation". In: *Sixteenth Annual Conference of the International Speech Communication Association*. 2015.

[8] J. H. Hansen and T. Hasan. "Speaker recognition by machines and humans: A tutorial review". In: *IEEE Signal processing magazine* 32.6 (2015), pp. 74–99.

[9] B. J. Lee et al. "Prediction of body mass index status from voice signals based on machine learning for automated medical applications". In: *Artificial intelligence in medicine* 58.1 (2013), pp. 51–61.

[10] V. Parsa and D. G. Jamieson. "Acoustic discrimination of pathological voice". In: *Journal of Speech, Language, and Hearing Research* (2001).

[11] C. Adnene, B. Lamia, and M. Mounir. "Analysis of pathological voices by speech processing". In: *Seventh International Symposium on Signal Processing and Its Applications, 2003. Proceedings.* Vol. 1. IEEE. 2003, pp. 365–367.

[12] E. S. Fonseca et al. "Wavelet time-frequency analysis and least squares support vector machines for the identification of voice disorders". In: *Computers in Biology and Medicine* 37.4 (2007), pp. 571–578.

[13] C. Sauder, M. Bretl, and T. Eadie. "Predicting voice disorder status from smoothed measures of cepstral peak prominence using Praat and Analysis of Dysphonia in Speech and Voice (ADSV)". In: *Journal of Voice* 31.5 (2017), pp. 557–566.

[14] K. P. Truong et al. "Arousal and valence prediction in spontaneous emotional speech: felt versus perceived emotion". In: (2009).

[15] C. R. Hodges-Simeon, S. J. Gaulin, and D. A. Puts. "Different vocal parameters predict perceptions of dominance and attractiveness". In: *Human Nature* 21.4 (2010), pp. 406–427.

[16] W. Zheng, J. Yu, and Y. Zou. "An experimental study of speech emotion recognition based on deep convolutional neural networks". In: *2015 international conference on affective computing and intelligent interaction (ACII)*. IEEE. 2015, pp. 827–831.

[17] A. M. Badshah et al. "Speech emotion recognition from spectrograms with deep convolutional neural network". In: *2017 international conference on platform technology and service (PlatCon)*. IEEE. 2017, pp. 1–5.

[18] R. Singh, D. Gencaga, and B. Raj. "Formant manipulations in voice disguise by mimicry". In: *2016 4th International Conference on Biometrics and Forensics (IWBF)*. IEEE. 2016, pp. 1–6.

[19] R. Singh, B. Raj, and D. Gencaga. "Forensic anthropometry from voice: an articulatory-phonetic approach". In: *2016 39th International Convention on*

Information and Communication Technology, Electronics and Microelectronics (MIPRO). IEEE. 2016, pp. 1375–1380.

[20] K. Franke and S. N. Srihari. "Computational forensics: An overview". In: *International Workshop on Computational Forensics*. Springer. 2008, pp. 1–10.

[21] C. Galea and R. A. Farrugia. "Forensic face photo-sketch recognition using a deep learning-based architecture". In: *IEEE Signal Processing Letters* 24.11 (2017), pp. 1586–1590.

[22] Z. Zhang. "Cause-effect relationship between vocal fold physiology and voice production in a three-dimensional phonation model". In: *The Journal of the Acoustical Society of America* 139.4 (2016), pp. 1493–1507.

[23] B. Gold, N. Morgan, and D. Ellis. *Speech and audio signal processing: processing and perception of speech and music*. John Wiley & Sons, 2011.

[24] W. Zhao, Y. Gao, and R. Singh. "Speaker identification from the sound of the human breath". In: *arXiv preprint arXiv:1712.00171* (2017).

[25] S. A. Memon et al. "Neural regression trees". In: *arXiv preprint arXiv:1810.00974* (2018).

[26] Y. Wen et al. "Disjoint mapping network for cross-modal matching of voices and faces". In: *arXiv preprint arXiv:1807.04836* (2018).

[27] H. Herzel et al. "Nonlinear dynamics of the voice: signal analysis and biomechanical modeling". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 5.1 (1995), pp. 30–34.

[28] C. Tao and J. J. Jiang. "Chaotic component obscured by strong periodicity in voice production system". In: *Physical Review E* 77.6 (2008), p. 061922.

Chapter 2

Target-Specific Models for Speaker Identification

This chapter presents target-specific models for voice-based forensic analysis of humans (VFAH). For illustrative purposes, we choose three specific tasks that are most relevant in VFAH: (1) speaker identification, (2) age estimation, and (3) height estimation. These three tasks are representative of typical target tasks in VFAH and have also been extensively researched, e.g., in [1, 2, 3]. We develop specialized models for each of the tasks. Nonetheless, these models can be easily extended to other VFAH tasks such as heart rate prediction, blood pressure prediction [4], emotion detection [5], etc. We start with the speaker identification task.

2.1 Generic Task-Specific Model Representation

The objective of target-specific modeling is to find a model $h \in \mathcal{H}$ of some class $\mathcal{H}$ that induces a deterministic map $h : \mathcal{X} \to \mathcal{Y}$ from some observed data domain $\mathcal{X}$ to the target domain $\mathcal{Y}$. In other words, given a data sample $x \in \mathcal{X}$, the model makes a prediction $\hat{y} = h(x)$. The target domain is given, and the target $y \in \mathcal{Y}$ can take continuous or discrete values.

Apparently, not any model h can make a correct prediction, and there could be an infinite number of h. Hence, we need a criterion to measure the "correctness" (or

error) of the prediction, i.e., we need a metric $c : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ satisfying the following requirements:

1. Non-negative: $c \geq 0$;
2. Reflective: for any y, $c(y, y) = 0$;
3. Symmetric (optional): for any y, $\hat{y}$, $c(y, \hat{y}) = c(\hat{y}, y)$;
4. Triangle inequality (optional): for any y, $\hat{y}_1$, $\hat{y}_2$, $c(y, \hat{y}_2) \leq c(y, \hat{y}_1) + c(\hat{y}_1, \hat{y}_2)$.

If only conditions 1 and 2 are satisfied (i.e., positive definiteness), c is a divergence that usually measures the distance from one probability distribution to another, such as Kullback–Leibler divergence and total variation distance. If additionally conditions 3 and 4 are satisfied, c is a pseudometric. If we further equip the metric with identity of indiscernibles, i.e., $c(x, y) = 0 \iff x = y$, the pseudometric becomes a metric. Common examples of metrics include Euclidean distance, metrics induced by norms, l_1 distance, Wasserstein distance between two probability measures, etc. A metric defines the "closeness" or "separation" of any two points in a given space (e.g., metric space, topological manifold). More profoundly, a metric can induce a topology that derives the concepts of continuity, convergence, and completeness. With a qualified metric, our objective is to determine an optimal model h^* that incurs the minimal error

$$h^* = \underset{h \in \mathcal{H}, (x,y) \in \mathcal{X} \times \mathcal{Y}}{\arg\min} c(y, h(x)) \tag{2.1}$$

2.2 Overview of Speaker Identification

Having defined a generic formulation for task-specific models, we move to a specific task—speaker identification from voices. Speaker identification is the task of determining the identity of a speaker from a voice sample. This is done by matching the given voice sample to samples within a database or a closed set of known speakers. The algorithms used for matching can be *text-dependent* with predefined spoken tokens [6], or *text-independent* with no prior knowledge of the spoken tokens [7]. Either type of

algorithm has been proven successful. Associated methodologies and technologies have also been widely applied for voice authentication in voice-password gated systems, banking fraud detection, multi-speaker tracking [8, 9], etc.

A standard speaker identification process involves three stages: (1) feature extraction, (2) speaker modeling, and (3) decision making. At the feature extraction stage, signal-based acoustic features such as Mel frequency cepstral coefficients (MFCC) are widely used in this task [10]. For speaker modeling, widely used statistical models include Gaussian mixture models (GMM) that incorporate a universal background model (UBM) as a reference (e.g., supervectors, i-vectors), probabilistic linear discriminant analysis (PLDA) based models, deep neural nets [8, 11, 12, 13], etc. For decision making, machine learning algorithms such as support vector machines (SVM), decision trees, random forests, etc., and deep learning models such as convolutional neural nets (CNN), deep belief nets, stacked auto-encoders, etc. [9] have been used.

A speaker identification system has two deployment phases: training and validation. During the training phase, the system enrolls the features for each candidate speaker, which may be registered in the form of sufficient statistics of statistical models or learned weights of neural nets. In the validation/testing phase, its decision-making algorithms match the features of the unknown speaker with those of enrolled speakers and make decisions based on some computed confidence measures, such as likelihood scores.

Despite their success, the effectiveness of these methodologies is limited in the context of VFAH for two reasons: (1) the length of the acquired utterance may be too short, and the number of samples too limited to make confident predictions with conventional statistical models or deep neural models; (2) the presence of background noise, channel distortions, and even voice disguise may mask the information present in the voice samples that may be relevant for profiling. For example, coast guard stations often receive "mayday" distress calls, some of which turn out to be hoax calls. Profiling from these recordings is challenging; even determining whether a call is a hoax or not is difficult because these calls are usually of very short duration and often have maritime noises in the background, such as the sound of the wind, boat

engines, etc. The problem that arises in such contexts is—how can we make accurate predictions about a speaker from such limited, noisy data with the possibility that the call may be in a disguised voice? Even if there is a prior pool of suspect recordings from earlier calls to match against, the problem is much less studied in the context of speaker identification.

Target-specific models address these problems through a judicious choice of feature representations best suited to the target task and a complimentary choice of models. With these, the problems mentioned above can be addressed to a great extent. For speaker identification, for example, we thus need (1) robust features to represent short speech signals that are resistant to change due to voice disguise or mimicry, easy to extract, and frequently appear enough in the voice signal, and (2) effective models to capture feature variability and to make accurate predictions.

To this end, we propose to use the sound of intervocalic breaths for speaker identification [14]. The advantages of using these are that breath sounds are ubiquitous, their intensity is measurable, their sound is not easy to disguise, and they carry physiological-structural signatures of the speaker's vocal tract, which are also not affected by voice disguise.

2.3 Speaker Identification from the Sound of Human Breath

Having introduced the general framework and speaker identification issues, we now examine the speaker identification potential of breath sounds in continuous speech. The goal is to demonstrate that breath sounds are indeed bio-signatures that can be used to identify speakers. We show that these sounds can yield remarkably accurate speaker recognition with appropriate feature representations and target-specific models.

2.3.1 Introduction

Speech is primarily produced during exhalation. In order to replenish the air in the lungs, speakers must periodically inhale. When inhalation occurs amid continuous speech, it is generally through the mouth. Intra-speech breathing behavior has been the subject of many studies, including patterns, cadence, and variations in energy levels. However, an often ignored characteristic is the *sound* produced during the inhalation phase of this cycle. Intra-speech inhalation is rapid and energetic, performed with an open mouth and glottis, effectively exposing the entire vocal tract to enable maximum air intake. This results in vocal tract resonances evoked by turbulence characteristic of the speaker's speech-producing apparatus. Consequently, inhalation sounds are expected to carry information about the speaker's identity. Moreover, unlike other spoken sounds, which are subject to active control, inhalation sounds are generally more natural and less affected by voluntary influences and exertions.

Intervocalic breath sounds are fundamentally different from relaxed breath sounds outside of speech. This is because breath plays a vital role in controlling the dynamics of speech. Natural speech is produced as a person exhales. It is almost impossible to produce sustained speech during inhalation [15]. As a person speaks, a specific volume of air is pushed out through the lungs and trachea into the vocal chambers, gated through the vocal folds in the glottis. Intervocalic breath sounds happen when the speaker exhausts the volume of air previously inhaled during a continuous speech and needs to inhale again. This inhalation is generally sharp, rapid, and volumetrically anticipatory of the following speech burst. The volume of air inhaled also depends on the air-intake capacity of the speaker's nasal and oral passageways, trachea, and inner structures leading to the lungs and further varies with myriad other factors related to the speaker's lung capacity, energy levels, muscular agility, etc.

Since exhalation is volumetrically linked to inhalation, the quality of the speech produced during exhalation also varies with all of these factors. Furthermore, when a person inhales, the vocal tract is usually lax and is in its natural shape. The lips are not protruded, nor do the articulators obstruct the vocal tract. In lax configurations,

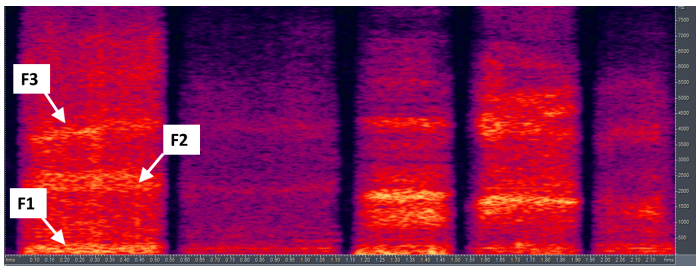
differences between speakers are expected to appear prominently as differences in resonant frequencies (due to differences in facial skeletal proportions and dimensions of the vocal chambers) and relative sound intensities (due to different lung capacities and states of health), etc.

For all these reasons, we expect many parameters of the speaker's persona to have their effects and possibly measurable signatures embedded in intervocalic breath sounds. Our goal in this paper is to experimentally show that these person-specific signatures within human breath (inhalation) sounds can be used for speaker identification.

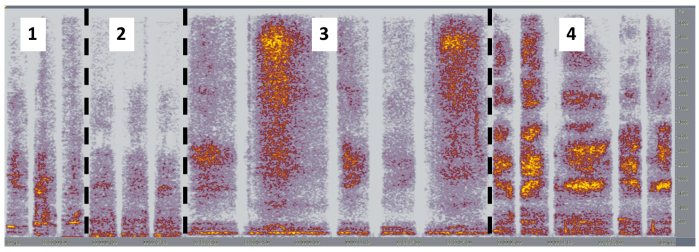
This can be useful in many real-life scenarios, especially those of forensic importance. For example, we have shown in an earlier study [16] that breath is invariant under disguise and impersonation. Its resonance patterns are usually not under the speaker's voluntary control and are extremely difficult to modify consistently for mechanical or cognitive reasons. Hence, the resonance patterns of intervocalic breath sounds are unique to speakers and visible in standard spectrographic representations of the speech signal. For example, Figure 2-1a and 2-1b show the spectrograms of the breath sounds of a child and four adult speakers, three of whom were attempting to impersonate the fourth speaker. This example is extracted from the public performances of voice artists attempting to impersonate the US presidential candidate in the 2016 elections in the USA—Mr. Donald Trump. We see the qualitative differences in the breath sounds in these examples. Even though, in reality, the impersonators of Mr. Trump sound very similar, their breath sounds show very distinctive speaker-specific patterns.

Related Work

Speaker identification from speech signals is widely applied and well-researched, with decades of work supporting it. Technology from this area has been the mainstay of forensic analysis of voice as well, an area that has been primarily centered around the topics of speaker identification [17, 18, 19, 20], verification [21, 22], detection of media tampering, enhancement of spoken content, and profiling [23, 24]. All of these areas have used articulometric considerations to their advantage. However, no reported studies strongly suggest using breath sounds in these forensic contexts. The closest



(a)



(b)

Figure 2-1: **(a)** Spectrogram of breath sounds of a four-year-old child during continuous speech. The formants F1, F2, and F3 correspond to the resonance of breath sounds and are clearly visible. **(b)** Breath sounds of Mr. Donald Trump (label 3) and his impersonators (labels 1, 2, and 4). All signals are energy-normalized and displayed on the same scale.

application comes from the medical field, where the sound of the patient's breath is used (very subjectively) by the clinician to deduce the patient's medical condition (such as lung function, respiratory diseases, response to their treatment, etc.) [25, 26, 27]. The stethoscope is the most ubiquitous medical instrument for this—i.e., for auscultation, the act of listening to the internal sounds of the body. Upon/after the publication of our work [14], several other works have also reported success in using breath sounds for speaker identification, such as [28, 29, 30].

2.3.2 Feature Formulations

Before building speaker identification models, we must ascertain that there is enough speaker-discriminatory information in breath sounds that can be successfully used for speaker identification. For this, we must develop appropriate feature representations that preserve, demonstrate, and magnify this speaker-discriminatory information.

Conventional MFCC-based temporal-spectral representation loses valuable informa-

tion in the original speech signal due to the irreversible steps taken in the computation process, such as triangular weighting and DCT. On the other hand, the i-vector is a widely accepted feature representation successfully deployed in state-of-art and commercial speaker identification and verification systems [31]. However, supervector and i-vector representations assume a multi-modal, or Gaussian mixture distribution of the speech spectrum [31, 32], which may not be appropriate or valid for turbulent sounds like breath. Moreover, these conventional methods face a significant performance drop for ultra-short (e.g., duration less than one second) speech signals, given their statistical nature [32, 33, 34]. Meanwhile, deep neural nets (DNNs) have demonstrated success in speaker identification with short utterances [35]. To verify the utility of different feature representations for breath sound, we compare i-vector features with a set of novel CNN-RNN-based features derived from constant-Q representations of the speech signal.

I-Vector Features

Identity-vector (i-vector) based feature representations are ubiquitously used in state-of-art speaker identification and verification systems. In order to obtain i-vectors for any speech recording, the distribution of Mel-frequency cepstral coefficient (MFCC) vectors derived from it is modeled as a Gaussian mixture. The parameters of this Gaussian mixture model (GMM) are, in turn, obtained through maximum *a posteriori* adaptation of a universal background model (UBM) that represents the distribution of all speech [36, 37]. The mean vectors of the Gaussian modes in the adapted GMM are concatenated into an extended vector, known as a GMM supervector, which represents the distribution of the MFCC vectors in the recording [12].

I-vectors are obtained through factor analysis of GMM supervectors. Following the factor analysis model, each GMM supervector $\mathbf{M}$ is modeled as $\mathbf{M} = \mathbf{m} + \mathbf{T}\mathbf{w}_{\mathbf{M}}$, where $\mathbf{m}$ is a global mean, $\mathbf{T}$ is a triangular loading matrix comprising bases representing a *total variability space*, and $\mathbf{w}_{\mathbf{M}}$ is the i-vector corresponding to $\mathbf{M}$. The loading matrix $\mathbf{T}$ and mean $\mathbf{m}$ are learned from training data through the Expectation-Maximization (EM) algorithm [38]. Subsequently, given any recording $\mathbf{M}$, its i-vector can also be

derived using EM.

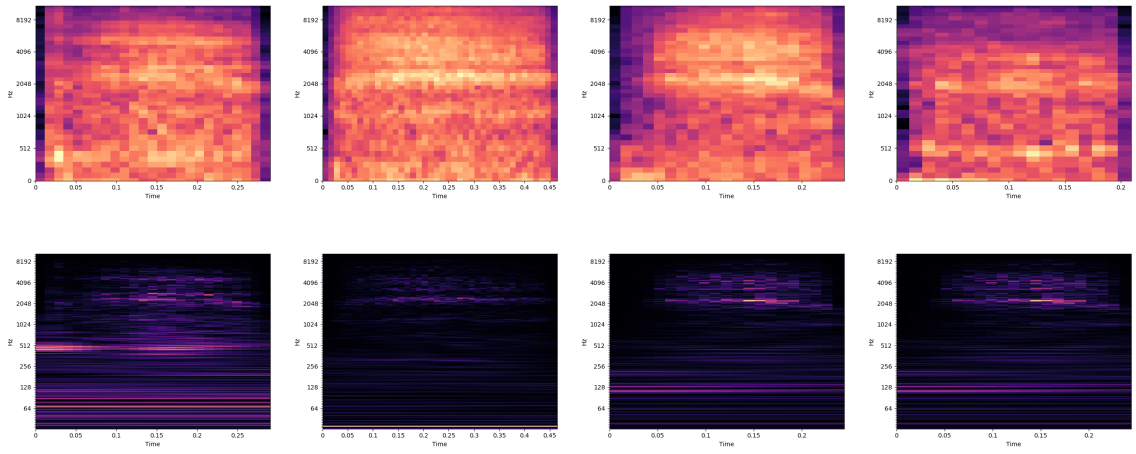
Constant-Q Spectrographic Features

Instead of conventional log-spectrograms or MFCC features, constant-Q representations of breath sounds may be derived. The constant-Q spectrum keeps the ratio of the center frequency to the filter bandwidth for each bin a constant w.r.t. the number of filters per octave [39]. The varied spacing of harmonics resulting from pitch variations on a normal spectrogram is converted to constant shifts in frequency on a constant-Q spectrogram. While this makes it robust and insensitive to pitch variations of the recordings from the same speaker, it also facilitates inter-speaker distinction. The filters used in constant-Q computation have geometrically spaced center frequencies and bandwidths like MFCC, but the transform does not need a further DCT step, and thus there is no further information loss. A counterpart to the constant-Q spectrogram is the Mel-scale spectrogram, which shares the property of logarithmic-spaced frequency bins. We compare these by showing the plots of Mel spectrograms and constant-Q spectrograms for the breath sounds from different speakers in Figure 2-2. We see that the constant-Q spectrograms show more distinct patterns than the Mel spectrograms, hence more suitable for representing breath sounds from different speakers with pitch variations.

Next, we show how a constant-Q spectrogram is obtained. Like the short-term Fourier transform (STFT), we first divide the speech signal into overlapping, consecutive frames. Then, we compute the constant-Q spectrum for each frame. These spectra are displayed in temporal succession, forming the spectrogram. As in general spectrographic representation used for speech signals, the frames span durations of $20 \sim 30$ ms and overlap by 50%–75%. For each frame $s[t]$, we compute its constant-Q transform $\mathbf{x}_t^{\mathrm{cq}}$ [40]. Specifically, the constant-Q transform of each frame $s[t]$ of the signal is given by

$$x^{\mathrm{cq}}[k] = \frac{1}{N_k} \sum_{n<N_k} s[n] w_{N_k}[n] e^{-j2\pi nQ/N_k}, \;\; k = 1, \ldots, K \tag{2.2}$$

with the window function $w[n]$, sampling frequency f_s, the minimum frequency f_0, the



(a) (b) (c) (d)

Figure 2-2: Mel-spectrograms (*Top*) and constant-Q spectrograms (*Bottom*) of the breath sounds from **(a)** female speaker 1, **(b)** female speaker 2, **(c)** male speaker 1, **(d)** male speaker 2.

maximum frequency $f_{\max}$, and the number of filters per octave b. Then the number of frequency bins $K = b \log_2 \frac{f_{\max}}{f_0}$, the k^{th} center frequency $f_k = f_0 2^{\frac{k}{b}}$, and the bandwidth of the k^{th} filter $\delta f_k^{\text{cq}} = f_k(2^{\frac{1}{b}} - 1)$. Therefore, the ratio of f_k to δf_k^{cq} is $Q = (2^{\frac{1}{b}} - 1)^{-1}$ (a constant), and the window length for the k^{th} bin $N_k = Q\frac{f_s}{f_k}$. Collecting all the K coefficients gives $\mathbf{x}_t^{\text{cq}} = [x^{\text{cq}}[1], \ldots, x^{\text{cq}}[K]]$. Concatenating $\mathbf{x}_t^{\text{cq}}$ for T frames into a matrix $\mathbf{X}$ gives us the constant-Q spectrogram for the input signal. We use these spectrograms as features for breath sounds.

The primary reason for the choice of constant-Q spectrograms is that the variations in the spacings of the harmonics due to variations in pitch on a normal spectrogram become constant shifts in frequency on a constant-Q spectrogram. In addition, the filters used in constant-Q computation have geometrically spaced center frequencies and bandwidths, like MFCC, allowing better discrimination for speech sounds in general.

2.3.3 Speaker Modeling via CNN-LSTM

In this section, we formulate the speaker identification problem and then describe a CNN-LSTM-based framework to solve the speaker identification problem.

Problem Formulation

Consider a collection of N constant-Q spectrograms $\mathbb{X} = \{\mathbf{X}_1, \ldots, \mathbf{X}_N\}$, $\mathbf{X}_i \in \mathbb{R}^{F \times T}$, $i = 1, \ldots, N$, where F is the number of frequency bins and T is the number of frames. We denote the collection of the corresponding labels as $\mathbb{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$, $\mathbf{y}_i \in \mathbb{R}^C$, $i = 1, \ldots, N$, where C is the number of speakers. The label $\mathbf{y}_i = \mathbf{1}_{i=c}$ for speaker class c has the c^{th} entry being 1 and the rest entries being 0s. Given the speech spectrograms $\mathbf{X}$, our goal is to design a classifier $h : \mathbb{R}^{F \times T} \to \mathbb{R}^C$ to predict the probability mass over the C classes: $\widehat{\mathbf{y}} = h(\mathbf{X}) = \mathbb{P}(\mathbf{y} \mid \mathbf{X})$. We first define the loss function

$$L(\widehat{\mathbf{y}}, \mathbf{y}) = D_{\mathrm{KL}}(\mathbf{y} \parallel \widehat{\mathbf{y}}) = \sum_{j=1}^{C} y_j \log \frac{y_j}{\widehat{y}_j} \tag{2.3}$$

where $D_{\mathrm{KL}}(\mathbf{y} \parallel \widehat{\mathbf{y}})$ is the Kullback–Leibler distance between the true probability mass $\mathbf{y}$ and the predicted probability mass $\widehat{\mathbf{y}}$. It is often used to measure the distance between densities. Now we define the classification risk

$$R(h) = \mathbb{E}_{(\mathbb{X},\mathbb{Y})\sim\mathcal{D}}\left[L(\widehat{\mathbf{y}}, \mathbf{y})\right] = \mathbb{E}_{(\mathbb{X},\mathbb{Y})\sim\mathcal{D}}\left[D_{\mathrm{KL}}(\mathbf{y} \parallel \widehat{\mathbf{y}})\right] \tag{2.4}$$

which is the expectation of the loss (2.3) over the data distribution $\mathcal{D}$. Since the true data distribution is unknown, we consider the empirical risk instead

$$\widehat{R}(h) = \tfrac{1}{N} \sum_{i=1}^{N} D_{\mathrm{KL}}(\mathbf{y}_i \parallel \widehat{\mathbf{y}}_i) = \tfrac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{C} y_{ij} \log \frac{y_{ij}}{\widehat{y}_{ij}} \tag{2.5}$$

which is the average of the losses over all the N samples. Hence, our optimization objective is to minimize the empirical risk (2.5)

$$h^* = \operatorname*{arg\,min}_{h \in \mathcal{H}} \widehat{R}(h) \tag{2.6}$$

where $\mathcal{H}$ is a class of classification functions. Later in this section, we construct $\mathcal{H}$ as a family of CNN-LSTM neural networks.

Returning to the generic task-specific model representation (2.1), the data representation (or data sample) $\mathbf{X}$ is the spectrogram, which is a two-dimensional real-valued

matrix. The target $\mathbf{y}$ is discrete-valued, which takes as many values as the number of different speakers. This is a classification task in machine learning terminology, where the classifier h is the CNN-LSTM neural model. A qualified error metric c is the Kullback-Leibler divergence. The empirical risk is minimized over a held-out training set using gradient descent via back-propagation.

Speaker Identification Model

Next, we describe the speaker identification model. With the constant-Q feature extracted from the intervocalic breath sounds, we propose a deep learning model—a convolutional neural net combined with a long short-term memory net (CNN-LSTM) model, to identify the speaker from breath sounds [14]. Figure 2-3 illustrates the proposed CNN-LSTM architecture. It comprises a convolutional layer, a max-pooling layer, an LSTM layer, a dropout layer, and a fully connected layer. The CNN part in the model learns shift and scale-invariant features from the constant-Q spectrogram. Following that, the LSTM part learns temporal connections of the CNN feature maps. Together they capture both the spatial and temporal information contained in the spectrogram while putting no constraint on the length of the input.

The convolutional layer convolves the same set of filters with an input spectrogram to learn its shift-variant features, which can be used to identify speakers. Specifically, for a spectrogram $\mathbf{X} \in \mathbb{R}^{F \times T}$, the convolutional layer convolves it with L filters $\mathbf{W}_i \in \mathbb{R}^{U \times V}$, $i = 1, \ldots, L$ with size (U, V). The resulting feature map $\mathbf{Z}_i^{\text{conv}} = \mathbf{X} * \mathbf{W}_i + b_i$, where b_i is a bias term. The feature map is then transformed $\mathbf{X}_i^{\text{conv}} = r(\mathbf{Z}_i^{\text{conv}})$ with a nonlinear activation function. We use the rectifier (or Rectified Linear Unit, ReLU) function $r(x) = \max(0, x)$.

The collection of feature maps $\mathbb{X}^{\text{conv}} = \{\mathbf{X}_1^{\text{conv}}, \ldots, \mathbf{X}_L^{\text{conv}}\}$ are then down-sampled along frequency with max-pooling mechanism [41] to reduce the amount of parameters and computation in the network, and hence to also control over-fitting. In order to learn the temporal information between the frames in each feature map, the down-sampled feature maps $\mathbb{X}^{\text{pool}}$ are flattened over time and fed into an LSTM layer. The LSTM layer consists of a sequence of memory units to selectively remember the past sequence

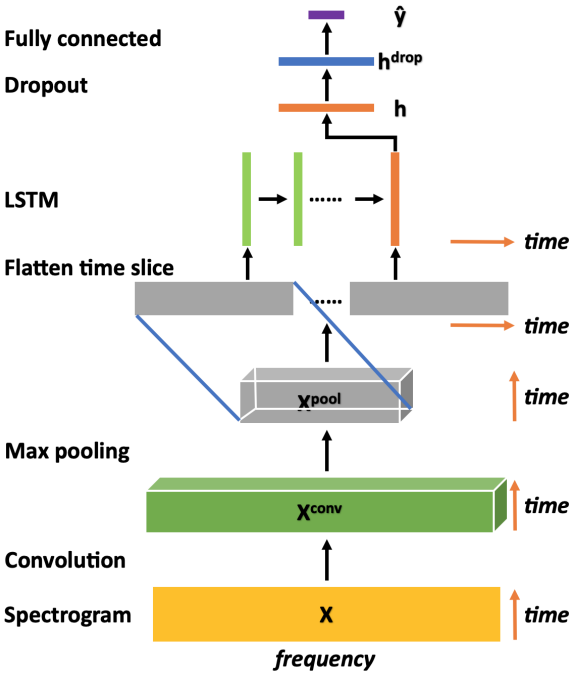


Figure 2-3: CNN-LSTM model architecture for speaker identification with breath sounds.

information. For a single unit, it has one memory cell and three control gates: the forget gate, the input gate, and the output gate [42]. We take the output from the last time step.

The output of the LSTM layer $\mathbf{h}$ is further fed into a dropout layer to avoid over-fitting [43]. The resultant output $\mathbf{h}^{\text{drop}}$ is finally passed to a fully connected layer and normalized onto a probability simplex using the soft-max function

$$\widehat{y}_i = \frac{e^{\mathbf{w}_i^T \mathbf{h}^{\text{drop}}}}{\sum_{j=1}^{C} e^{\mathbf{w}_j^T \mathbf{h}^{\text{drop}}}}, \ i = 1, \ldots, C \tag{2.7}$$

where $\mathbf{w}_i$ is the weight in the fully connected layer. This final output $\widehat{\mathbf{y}} = [\widehat{y}_1, \ldots, \widehat{y}_C]$ comprises the multi-class likelihoods for the C speakers. We can then minimize the objective (2.6), where the function class $\mathcal{H} = \{h_{\mathbf{w}}\}$ for all the parametrization of $\mathbf{w}$ in

the network. Substitution (2.7) into (2.6) yields

$$\begin{aligned}\mathbf{w}^* &= \arg\min_{\mathbf{w}} \widehat{R}(h_{\mathbf{w}}) \\ &= \arg\min_{\mathbf{w}} \tfrac{1}{N} \sum_{i=1}^{N} (\log \sum_{j=1}^{C} e^{\mathbf{w}_j^T \mathbf{h}^{\text{drop}}} - \mathbf{w}_i^T \mathbf{h}^{\text{drop}})\end{aligned} \tag{2.8}$$

To obtain the optimal set of parameters $\mathbf{w}^*$, we train our network using back-propagation. The gradients of the risk with regard to the parameters in the last layer are back-propagated, and the parameters are updated until convergence.

2.3.4 Experiments

In this section, we describe our experimental setup and compare the results of our CNN-LSTM speaker identification framework with the results of i-vectors.

Task and Data

The data used for both sets of experiments are the same. We select the LDC Hub-4 1997 Broadcast News database [44], comprising single-channel recordings of reading speeches from multiple news anchors and people interviewed within the news episodes. The recordings are sampled at 16000 Hz. We use Sphinx-3, a state-of-art Hidden Markov Model (HMM) based automatic speech recognition (ASR) system [45, 46], to obtain accurate phoneme segmentation for all the speech signals in the database. The ASR system is trained on this database, and the resultant acoustic models are used to obtain highly accurate phoneme segmentation. Breath is modeled as a phoneme during the training process; thus, the process of phoneme segmentation directly yields the breath sounds that we need for our experiments. Each breath segment's duration is less than one second, and we do not concatenate them. We have verified that these speech segments are indeed breath sounds rather than silences. The complete set of breath sounds extracted from the Hub-4 database includes more than 3000 combinations of the speaker, channel (broadband and telephone), fidelity (high, low, medium), and type of speech (read and conversational). Since this study aims to demonstrate that

breath can be used to identify speakers, we do not attempt to explore the performance in different speech styles, physical conditions, channel types, noise conditions, etc. Hence, we only choose breath sounds corresponding to high-fidelity clean reading speech signals for our experiments, resulting in a subset consisting of 50 speakers. The breath data consists of 9915 instances corresponding to 50 speakers, which is sufficient for our experiments.

For comparison, we construct a baseline speaker identification system using a standard i-vector-based approach and all utterances (not just breath sounds). We also compare speaker identification using the /EY/ phoneme (as in "mayday"). The /EY/ data consists of 10400 samples corresponding to 50 speakers. Next, we describe the experiments in detail.

I-Vector Based Experiments

The baseline speaker identification system adopts a standard text-independent i-vector-PLDA-based approach [47]. For each speaker, 70% of the data is used as the training set and 30% as the test set. For an utterance, we use a 20 ms window with 10 ms overlap and short time Gaussianization with 3 sec sliding window [47], and extract the 20-dimensional (including energy) MFCC features with delta and double delta coefficients, resulting in 60-dimensional vectors. We train a universal background model (UBM) of 512 Gaussian mixtures on the training data and generate 400-dimensional i-vectors. After length normalization and whitening [48], we perform PLDA for scoring. The Kaldi toolkit is used for this process [49].

We use similar settings for i-vector experiments with breath and /EY/ sounds, except that we compute i-vectors of different dimensions (20, 30, 40, 60, 80, 100, 200, 300). Besides the i-vector-PLDA pipeline, we also compare with classifying i-vectors using SVM and neural nets. For SVM experiments, we implement a multi-class SVM using the LIB-SVM library [50]. We reduce the dimension of i-vectors greater than 60 to 49 using Linear Discriminant Analysis (LDA) [51]. The neural networks are in the form of multi-layer perceptrons and implemented using Theano [52] and Keras [53] toolkits. The network architecture includes different activation functions in different

layers: the first hidden layer uses the rectified linear unit (ReLU), the second hidden layer, when present, uses the sigmoid function, and the output layer uses the soft-max function. The learning rate is 0.1 with a decay rate of $1e^{-9}$ and a momentum of 0.9. The batch size used in each epoch of training is 400.

The speaker identification accuracies obtained under different settings are shown in Table 2.1 and Figure 2-4a. We see that the i-vector-PLDA pipeline on all utterances (row 1) achieves the best accuracy. The performance of i-vector-based approaches significantly degrades as the utterance duration reduces to less than one second (row 2–3, 5–9). This is because ultra-short utterances fail to satisfy the statistical assumptions of i-vector-based approaches. In comparison, the i-vector with /EY/ sound (row 2–3) yields better performance than the i-vector with breath sound (row 5–9). This is because the /EY/ sound has distinct formants; hence, its spectral distribution is more Gaussian-like and more amenable to GMM modeling. The PLDA scoring (row 2) achieves higher accuracy than the SVM classification (row 3) and further supports this observation.

On the other hand, PLDA scoring (row 5) performs relatively worse than SVM classification (row 7) for breath sounds. This is because the breath sound is turbulent, less Gaussian-like, not context-dependent, less prone to external influences, and has less session variability, rendering the subspace decomposition assumption less valid. SVM classification with dimensionality reduction (row 7) achieves higher accuracy than SVM without dimensionality reduction (row 6). This is because SVM does not operate well in high-dimensional spaces. This is supported by the observation that neural nets (row 8) achieve higher accuracy than SVMs (row 6) in high dimensions, as also demonstrated in Figure 2-4a. Further, neural nets with one layer (row 8) yield better performance than two-layer neural nets (row 9). These results suggest that the speaker-specific information is supported in a lower-dimensional quasi-linear subspace. Hence, the i-vector with LDA yields a more robust and consistent performance across dimensions.

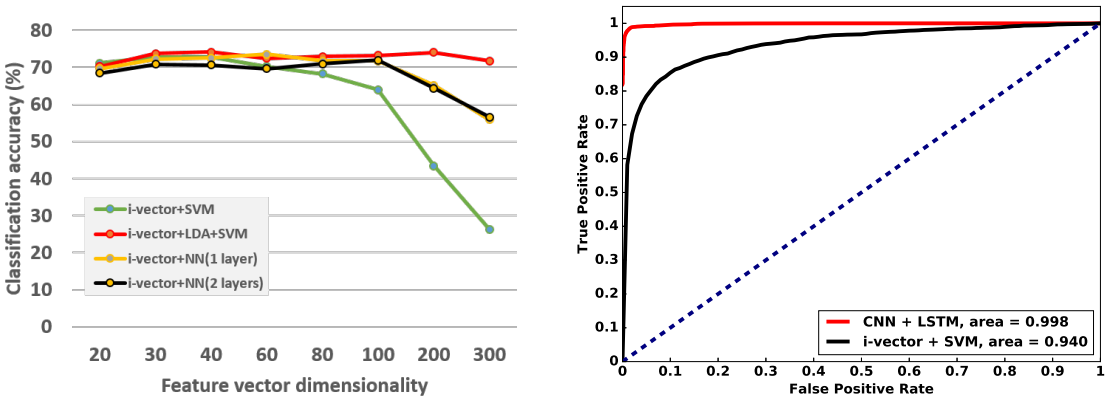
Table 2.1: Speaker Identification Results

	Method	Accuracy (%)
1	all+ivec-plda	**96.3**
2	ey+ivec-plda	81.8
3	ey+ivec-lda-svm	76.9
4	ey+cnn-lstm	**90.7**
5	breath+ivec-plda	73.4
6	breath+ivec-svm	72.8
7	breath+ivec-lda-svm	74.1
8	breath+ivec-1nn	73.5
9	breath+ivec-2nn	71.9
10	breath+cnn-lstm	**91.3**

CNN-LSTM Experiments

For CNN-LSTM-based experiments, we first convert all breath sounds to constant-Q spectrograms [40]. We use 48 filters per octave, sampling frequency $f_s = 44100$ Hz, lowest frequency $f_{min} = 27.5$ Hz, and highest frequency $f_{max} = f_s/2$, resulting in 463 frequency bins for each frame. To compensate for pitch variations within speakers, we further augment the data using the elastic transform [54] with $\sigma = 2$ and $\alpha = 15$. Then, we select the network hyper-parameters using cross-validation to avoid over-fitting. For each speaker, we select 70% of the utterances as the training set, 20% as the validation set, and 10% as the test set. We configure the network to have the following parameters—input dimension: $463 \times T$ (T represents varying time duration), CNN filter size: $8 \times 3 \times 3$, max pooling stride: 2×1, dropout rate: 0.4, output dimension: 44. The network is implemented using Theano [52], and trained using Adadelta [55] with decay constant 0.9 and batch size 1. The training is stopped when the error on the validation set stops decreasing.

Table 2.1 and Figure 2-5 show the speaker identification performance using breath and /EY/ sounds. Our CNN-LSTM model with constant-Q features (row 4, 10) achieves remarkably higher accuracy than i-vector-based methods on ultra-short utterances. This suggests that constant-Q spectrograms do not depend on spectral distribution assumptions and are more distinctive and robust feature representations



(a) (b)

Figure 2-4: Speaker identification performance using breath sounds. **(a)** Speaker identification accuracy with the change of i-vector dimensionality in four different classifier settings. **(b)** ROC curves for CNN-LSTM and i-vector-SVM.

for short speech signals than i-vectors. Further, our CNN-LSTM model extracts robust spatial-temporal features from constant-Q representations and performs better than conventional models. Moreover, CNN-LSTM with breath sounds (row 10) gives higher accuracy than with /EY/ sounds (row 4) and a larger area under the curve (AUC) in the receiver operating curves (ROC), as shown in Figure 2-5. This validates our hypothesis that intervocalic breath sounds carry unique and speaker-characteristic information and, when properly represented, can effectively and robustly identify speakers.

As a final test, we use the individual breath recordings in a speaker verification setting. For each speaker, a two-class classifier is trained to distinguish between the speaker and all other speakers. Figure 2-4b shows the ROC for both the CNN-LSTM based experiments and the i-vector based experiments. Our CNN-LSTM framework achieves a higher true positive rate than the i-vector-based approach. In both cases, the large AUC sufficiently demonstrates the potential of using breath sounds in practical speaker identification tasks.

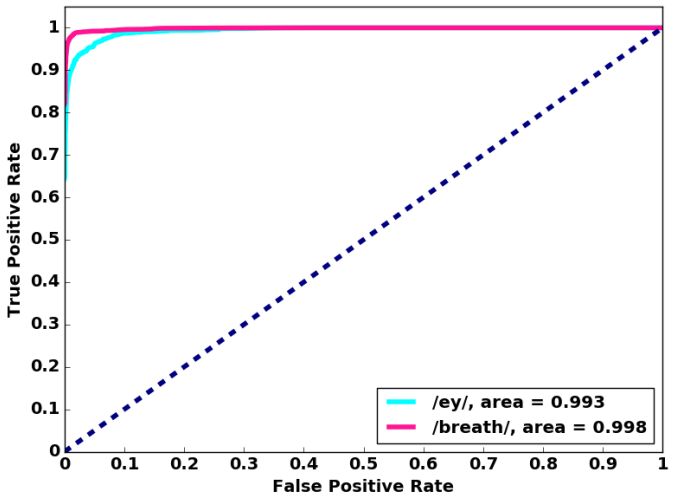


Figure 2-5: Speaker identification performance for breath and /EY/ sounds using CNN-LSTM framework.

2.3.5 Conclusions

Experiments with both i-vectors and constant-Q spectrograms show that breath sounds can be successfully used to identify speakers. The accuracies are surprisingly good for the clean speech signals we used in our experiments. The proposed constant-Q representations for breath sounds can effectively preserve the invariant speaker-discriminatory information under other confounding influences (e.g., noise, disguise, impersonation) and maximize the cross-speaker distinctions. The proposed CNN-LSTM model achieves high speaker identification performance. It automatically learns the shift-invariant and temporal features and combines feature extraction, speaker modeling, and decision making into a single pipeline. This model is also distribution-assumption free and works effectively for short recordings.

Note that since our primary objective in this study is to demonstrate that the sound of the human breath can be used for speaker identification, this choice of features was judiciously made to fulfill our goal of providing proof of concept. In the future, we can explore the ability of other phonemes to identify speakers individually and collectively and evaluate other feature representations and model choices. We can also extend the work to investigate these under adverse conditions such as noise and disguise.

References

[1] P. Rose. *Forensic speaker identification*. cRc Press, 2002.

[2] R. Singh et al. "The relationship of voice onset time and voice offset time to physical age". In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2016, pp. 5390–5394.

[3] R. Singh, B. Raj, and D. Gencaga. "Forensic anthropometry from voice: an articulatory-phonetic approach". In: *2016 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. IEEE. 2016, pp. 1375–1380.

[4] B. Schuller, F. Friedmann, and F. Eyben. "Automatic recognition of physiological parameters in the human voice: Heart rate and skin conductance". In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE. 2013, pp. 7219–7223.

[5] M. Tahon, G. Degottex, and L. Devillers. "Usual voice quality features and glottal features for emotional valence detection". In: *Speech Prosody 2012*. 2012.

[6] T. Dutta. "Text dependent speaker identification based on spectrograms". In: *Proceedings of Image and vision computing* (2007), pp. 238–243.

[7] H. Gish and M. Schmidt. "Text-independent speaker identification". In: *IEEE signal processing magazine* 11.4 (1994), pp. 18–32.

[8] R. Togneri and D. Pullella. "An overview of speaker identification: Accuracy and robustness issues". In: *IEEE circuits and systems magazine* 11.2 (2011), pp. 23–61.

[9] S. S. Tirumala and S. R. Shahamiri. "A review on Deep Learning approaches in Speaker Identification". In: *Proceedings of the 8th international conference on signal processing systems*. ACM. 2016, pp. 142–147.

[10] V. Tiwari. "MFCC and its applications in speaker recognition". In: *International journal on emerging technologies* 1.1 (2010), pp. 19–22.

[11] J. H. Hansen and T. Hasan. “Speaker recognition by machines and humans: A tutorial review”. In: *IEEE Signal processing magazine* 32.6 (2015), pp. 74–99.

[12] T. Kinnunen and H. Li. “An overview of text-independent speaker recognition: From features to supervectors”. In: *Speech communication* 52.1 (2010), pp. 12–40.

[13] R. Jahangir et al. “Text-independent speaker identification through feature fusion and deep neural network”. In: *IEEE Access* 8 (2020), pp. 32187–32202.

[14] W. Zhao, Y. Gao, and R. Singh. “Speaker identification from the sound of the human breath”. In: *arXiv preprint arXiv:1712.00171* (2017).

[15] D. W. Warren. “Aerodynamics of speech production”. In: *Contemporary issues in experimental phonetics* 30 (1976), pp. 105–137.

[16] R. Singh, D. Gencaga, and B. Raj. “Formant manipulations in voice disguise by mimicry”. In: *4th International Workshop on Biometrics and Forensics (IWBF)*. IEEE. Limassol, Cyprus, 2016.

[17] D. Meuwly. “Forensic speaker recognition”. In: *Wiley Encyclopedia of Forensic Science* (2009).

[18] C. Champod and D. Meuwly. “The inference of identity in forensic speaker recognition”. In: *Speech communication* 31.2 (2000), pp. 193–203.

[19] P. Rose. *Forensic speaker identification*. London: CRC Press, 2003.

[20] J. P. Campbell et al. “Forensic speaker recognition”. In: Institute of Electrical and Electronics Engineers. 2009.

[21] T. Becker, M. Jessen, and C. Grigoras. “Forensic speaker verification using formant features and Gaussian mixture models”. In: *Interspeech*. 2008, pp. 1505–1508.

[22] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn. “Speaker verification using adapted Gaussian mixture models”. In: *Digital signal processing* 10.1 (2000), pp. 19–41.

[23] R. Singh, J. Keshet, and E. Hovy. “Profiling hoax callers”. In: *IEEE International Symposium on Technologies for Homeland Security*. Waltham, USA, 2016.

[24] R. Singh, B. Raj, and D. Gencaga. "Forensic anthropometry from voice: an articulatory-phonetic approach". In: *Biometrics & Forensics & De-identification and Privacy Protection, MIPRO2016*. Croatia. 2016.

[25] R. G. Loudon, L. Lee, and B. J. Holcomb. "Volumes and breathing patterns during speech in healthy and asthmatic subjects". In: *Journal of Speech, Language, and Hearing Research* 31.2 (1988), pp. 219–227.

[26] A. Henderson, F. Goldman-Eisler, and A. Skarbek. "Temporal patterns of cognitive activity and breath control in speech". In: *Language and Speech* 8.4 (1965), pp. 236–242.

[27] A. L. Winkworth et al. "Breathing patterns during spontaneous speech". In: *Journal of Speech, Language, and Hearing Research* 38.1 (1995), pp. 124–144.

[28] L. Lu et al. "I sense you by breath: Speaker recognition via breath biometrics". In: *IEEE Transactions on Dependable and Secure Computing* 17.2 (2017), pp. 306–319.

[29] V.-T. Tran and W.-H. Tsai. "Stethoscope-sensed speech and breath-sounds for person identification with sparse training data". In: *IEEE Sensors Journal* 20.2 (2019), pp. 848–859.

[30] A. İlerialkan, A. Temizel, and H. Hacıhabiboglu. "Speaker and posture classification using instantaneous intraspeech breathing features". In: *arXiv preprint arXiv:2005.12230* (2020).

[31] N. Dehak et al. "Front-end factor analysis for speaker verification". In: *IEEE Transactions on Audio, Speech and Language Processing* 19.4 (2011), pp. 788–798.

[32] P. Kenny et al. "JFA-based front ends for speaker recognition". In: *2014 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE. 2014, pp. 1705–1709.

[33] A. Kanagasundaram et al. "I-vector based speaker recognition on short utterances". In: *Proceedings of the 12th Annual Conference of the International Speech*

Communication Association. International Speech Communication Association. 2011, pp. 2341–2344.

[34] P. Kenny et al. "PLDA for speaker verification with utterances of arbitrary duration". In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE. 2013, pp. 7649–7653.

[35] J. Rohdin et al. "End-to-end DNN based speaker recognition inspired by i-vector and PLDA". In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2018, pp. 4874–4878.

[36] D. A. Reynolds. "Comparison of background normalization methods for text-independent speaker verification". In: *Eurospeech*. 1997.

[37] D. Reynolds. "Universal background models". In: *Encyclopedia of biometrics* (2015), pp. 1547–1550.

[38] T. K. Moon. "The expectation-maximization algorithm". In: *IEEE Signal processing magazine* 13.6 (1996), pp. 47–60.

[39] J. C. Brown and M. S. Puckette. "An efficient algorithm for the calculation of a constant Q transform". In: *The Journal of the Acoustical Society of America* 92.5 (1992), pp. 2698–2701.

[40] J. C. Brown. "Calculation of a constant Q spectral transform". In: *The Journal of the Acoustical Society of America* 89.1 (1991), pp. 425–434.

[41] M. D. Zeiler and R. Fergus. "Visualizing and understanding convolutional networks". In: *European conference on computer vision*. Springer. 2014, pp. 818–833.

[42] S. Hochreiter and J. Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[43] N. Srivastava et al. "Dropout: a simple way to prevent neural networks from overfitting". In: *Journal of Machine Learning Research* 15.1 (2014), pp. 1929–1958.

[44] J. Fiscus et al. “1997 English Broadcast News Speech (HUB4)”. In: *LDC98S71*. Linguistic Data Consortium. USA, 1998.

[45] K. Seymore et al. “The 1997 CMU Sphinx-3 English broadcast news transcription system”. In: *DARPA Broadcast News Transcription and Understanding Workshop*. Citeseer. 1998.

[46] J. M. Huerta, S. Chen, and R. M. Stern. “The 1998 carnegie mellon university sphinx-3 spanish broadcast news transcription system”. In: *Proc. of the DARPA Broadcast News Transcription and Understanding Workshop*. Citeseer. 1999.

[47] Y. Jiang et al. “PLDA modeling in i-vector and supervector space for speaker verification”. In: *Annual Conference of the International Speech Communication Association (Interspeech)*. 2012.

[48] D. Garcia-Romero and C. Y. Espy-Wilson. “Analysis of i-vector length normalization in speaker recognition systems”. In: *Twelfth annual conference of the international speech communication association*. 2011, pp. 249–252.

[49] D. Povey et al. “The Kaldi speech recognition toolkit”. In: *IEEE 2011 workshop on automatic speech recognition and understanding*. CONF. IEEE Signal Processing Society. 2011.

[50] C.-C. Chang and C.-J. Lin. “LIBSVM: a library for support vector machines”. In: *ACM Transactions on Intelligent Systems and Technology* 2.27 (2011), pp. 1–27.

[51] A. J. Izenman. “Linear discriminant analysis”. In: *Modern multivariate statistical techniques*. Springer, 2013, pp. 237–280.

[52] Theano Development Team. “Theano: A Python framework for fast computation of mathematical expressions”. In: *arXiv e-prints* abs/1605.02688 (May 2016). URL: `http://arxiv.org/abs/1605.02688`.

[53] F. Chollet. *Keras*. `https://github.com/fchollet/keras`. 2015.

[54] P. Y. Simard, D. Steinkraus, and J. C. Platt. “Best practices for convolutional neural networks applied to visual document analysis”. In: *Proceedings of the Seventh International Conference on Document Analysis and Recognition*. Vol. 3. 2003, pp. 958–962.

[55] M. D. Zeiler. “ADADELTA: an adaptive learning rate method”. In: *arXiv preprint arXiv:1212.5701* (2012).

Chapter 3

Target-Specific Models for Age and Height Estimation

In this chapter, we continue to present target-specific models for voice-based forensic analysis of humans (VFAH). In the previous chapter, our focus was on the *feature*—we demonstrated how the choice of more targeted features can improve performance using the example task of speaker identification. In this chapter and the next, our focus is more on the *models*. We investigate the effects of using better features *while* refining the models to be more specific to the target task. As examples, we focus on two challenging VFAH tasks: age and height estimation. In this chapter, we first discuss direct modeling approaches for these via regression. Next, we introduce an indirect modeling approach—regression-via-classification and propose a new model for this: the Neural Regression Tree (NRT).

3.1 Direct Modeling Approaches

Age and height estimation are treated differently from speaker identification because the target y is now continuous-valued. This represents a continuous-valued prediction or *regression* task in machine learning terminology. For such tasks, direct approaches employ traditional regression models such as linear regression, ridge regression, least absolute shrinkage, and selection operator (LASSO) regression, or deep learning models

such as multi-layer perceptron (MLP), convolutional neural network (CNN), etc. One of the many qualified metrics used in these approaches is the l_2 distance.

In the context of age estimation, commonly used features include fundamental frequency, Mel-frequency cepstral coefficients (MFCC), supervectors, i-vectors, etc. Prediction models that work well with these features include support vector machine (SVM), Gaussian mixture models (GMM), neural networks, etc [1, 2, 3, 4, 5].

In the context of height estimation, commonly used features include MFCC, linear predictive coding (LPC) coefficients, fundamental frequency, harmonic-to-noise ratio, subglottal resonance, i-vectors, etc., and the corresponding models include SVM, linear regression, GMM, Gaussian process, non-negative factor analysis, neural networks, etc [6, 7, 8, 9, 10, 11].

However, the relationship between the features and the target variable (e.g., age or height) is generally unknown and may not be deterministic. The general approach to the problem is to assume a formula for the relationship and estimate the formula's details from training data. Linear regression models assume a linear relationship between the features and the target. Other models, such as neural networks, assume a non-linear relationship. The problem is that the model parameters for one data regime may not be appropriate for others. Statistical fits of the model to the data will minimize a measure of the overall prediction error but may not be truly appropriate for any specific subset of the data. For instance, age or height prediction involves a trade-off between selecting a subset of features correlated highly with age or height and prediction accuracy. The prediction error may be high for the young or the old, with different levels of variability for different subsets of speech features. Thus, merely fitting a single model to the data may minimize the overall prediction error but may not be appropriate for any specific subset of the data.

Therefore, the problem that arises is this: how can we construct a model that can adaptively optimize its prediction performance over subsets of data to accurately estimate age or height across an entire range of speakers? To address the above problem, we need to (1) devise a model that can predict based on local decisions made on optimally partitioned data and (2) derive local features for optimal prediction.

There are two solutions to these requirements. One is partitioning the input (feature) space and making predictions by combining local predictions. Many models utilize such a strategy. Kernel regressions, for example, predict using inherited data "closeness" information exploited by local kernels, such as Gaussian kernel, polynomial kernel [12], Mercer kernel [13], locally adaptive kernel [14], etc. Another example is regression trees, which predict using hierarchical local predictors. They include decision trees such as CART [15], ID3 [16], m5 [17], C4.5 [18], ensemble random forests [19], and more recent models that combine the power of tree structures and neural nets, such as convolutional decision trees [20], neural decision trees [21, 22], adaptive neural trees [23], deep neural decision forests [24], and deep regression forests [25], to name a few. All of these approaches partition the input feature space.

Notwithstanding their merits, though, they share two issues: first, for high-dimensional data, any computed partition runs the risk of over-fitting; second, there is no guarantee that the data is partitioned in a way to maximize prediction performance [16, 26].

3.2 Indirect Modeling Approaches

Another solution, which is the path we take, is partitioning the output (target) space, i.e., partitioning based on the target variable. Formally, given a response variable y that takes continuous values in the range $(y_{\min}, y_{\max})$, we find a set of thresholds $t_0, \ldots, t_N$, and map the response variable into bins as $y \mapsto C_n$ if $t_{n-1} < y \leq t_n$ for $n = 1, \ldots, N$. This effectively defines a partition Π on a set $\mathcal{Y} \subset \mathbb{R}$ as $\Pi(\mathcal{Y}) \coloneqq \{C_1, \ldots, C_N\}$ satisfying $\bigcup_{n=1}^{N} C_n = \mathcal{Y}$. The bins C_n are mutually disjoint. This process, which essentially converts the continuous-valued y into a categorical one C, is referred to as *discretization*. In order to determine the value y for any x, we must find out which bin C_n the feature x belongs to. Once the appropriate bin has been identified, the estimated y can be computed via local regression within the bin C_n. Consequently, the problem of regression is transformed into classification. This process of converting a regression problem to a classification problem is known as regression-via-classification (RvC) [27]. The idea for RvC was introduced in [28], in which k-means clustering was

employed to categorize numerical variables.

3.2.1 Regression-via-Classification

Regression-via-Classification (RvC) is the process of converting a regression problem to a classification one. The RvC framework permits partitioning the input space based on output variables that maximize the local predictors' performance and optimize the overall performance. The challenge is to decide on an optimal discretization strategy for the response variable y. Conventional approaches usually discretize y based on equally probable intervals, i.e., intervals with the same number of elements, or equal-width intervals, i.e., intervals of the same range [27, 29]. However, these approaches are *ad-hoc*.

Further, a naïve implementation of RvC can result in very poor regression. Inappropriate choice of bin boundaries $\{t_i\}$ can result in bins that are too difficult to classify (since classification accuracy depends on the distribution of feature x within the bins). Although permitting near-perfect classification, the bins may be too wide, and the corresponding "regression" may be meaningless.

To address the problem of deciding the optimal discretization within RvC, we propose a tree-structured RvC model with neural node-classifiers to learn the optimal partition thresholds [30]. We refer to this framework as a "neural regression tree" (NRT).

3.2.2 Neural Regression Tree

Inspired by the original idea of regression trees [16], we follow a greedy strategy of hierarchical binary partitioning of the target variable y, where each split is locally optimized. This results in a tree-structured RvC model with a classifier at each node. Moreover, such a model structure affords us additional optimization. Instead of using a single generic feature for classification (such as margin-based linear classifiers), we can now optimize the features extracted from the data individually for each classifier in the tree. We employ neural node classifiers to partition the data and optimize the

local features at each node. At the leaf nodes, we can then use the expected mean of local bins as the final predicted value of the target variable.

Ideally, the NRT must optimize the discretization boundaries for classification and regression accuracy. However, the complexity of joint optimization over the discretization boundaries $\{t_n\}$ and the classifier parameters $\{\theta_n\}$ scales exponentially with n. To solve this problem, we adopt a divide-and-conquer strategy and perform greedy optimization over each node-classifier. We further propose a triviality loss to regularize the node optimization.

Next, we formulate the NRT model for the optimal discretization of the target variables in RvC and provide the algorithm to solve the model.

Partition

The key aspect of an RvC system is its partition method. We define the partition Π on a set $\mathbb{Y} \subset \mathbb{R}$ as

$$\Pi(\mathbb{Y}) = \{C_1, \ldots, C_N\} \tag{3.1}$$

satisfying the requirement that $\bigcup_{n=1}^{N} C_n = \mathbb{Y}$ and C_ns are mutually disjoint. When acting on a $y \in \mathbb{Y}$, $\Pi(y) \coloneqq C_n$ subjected to $y \in C_n$.

NRT Model Formulation

Formally, an RvC framework consists of two main rules: a classification rule and a regression rule. The classification rule classifies an input x into disjoint bins, i.e., $h_\theta : x \mapsto \{C_1, \cdots, C_N\}$ with parameter θ (θ, for example, could be the parameter of a specific classifier, such as a neural net or an SVM), where $C_n = \Pi(y)$ corresponds to $t_{n-1} < y \leq t_n$ for $n = 1, \ldots, N$. The regression rule $r : (x, C_n) \mapsto (t_{n-1}, t_n]$ maps the combination of input x and bin C_n onto the interval $(t_{n-1}, t_n]$ which contains the prediction $\hat{y}(x)$) Then, the combined RvC rule that predicts the value of the target variable for an input x is

$$\hat{y}(x) = r(x, h_\theta(x)) \tag{3.2}$$

Instead of making a hard assignment of bins, alternatively, the classification rule h_θ may make a "soft" assignment by mapping an input x onto the N-dimensional probability simplex, i.e., $h_\theta : x \mapsto \mathbb{P}_N$ where $\mathbb{P}_N$ represents the set of N-dimensional non-negative vectors whose entries sum to 1. The output of this classification rule is, therefore, the vector of *a posteriori* probabilities over the N classes (bins), i.e., $h_\theta^n(x) = P(C_n \mid x)$ where $h_\theta^n(x)$ represents the n^{th} component of $h_\theta(x)$. Hence, the probabilistic RvC rule is given by

$$\hat{y}(x) = \mathbb{E}_{C_n}\left[r(x, C_n)\right] = \sum_{n=1}^{N} h_\theta^n(x) r_{C_n}(x) \tag{3.3}$$

where $r_{C_n}(\cdot) \coloneqq r(\cdot, C_n)$ fixes the second coordinate of r at C_n. Defining an error function $\mathcal{E}(y, \hat{y}(x))$ between the true y and the estimated $\hat{y}(x)$, our objective is to determine the partition thresholds $\{t_0, \ldots, t_N\}$ and the classifier parameters $\{\theta_0, \ldots, \theta_N\}$ such that the expected error is minimized

$$\{t_n^*\}, \{\theta_n^*\} \leftarrow \arg\min_{t,\theta} \mathbb{E}_x\left[\mathcal{E}(y, \hat{y}(x))\right] \tag{3.4}$$

Note that the number of thresholds $(N + 1)$ is also a variable that may be manually set or explicitly optimized. In practice, instead of minimizing the expected error, we minimize the empirical error $\text{avg}(\mathcal{E}(y_i, \hat{y}(x_i)))$ computed over a training set.

However, joint optimization of $\{t_n\}$ and $\{\theta_n\}$ is a challenging problem as it scales exponentially with n. To solve this problem, we adopt a divide-and-conquer approach and propose a binary tree-based algorithm—the neural regression tree (NRT) mentioned earlier—to solve the RvC problem, where each node in the tree is greedily optimized. The structure of the proposed binary tree is shown in Figure 3-1.

We now describe the binary tree algorithm. For notational convenience, the nodes are numbered such that for two nodes n_i and n_j, if $i < j$, n_i occurs either to the left of n_j or above it in the tree. Each node n in the tree is associated with a threshold t_n, which is used to partition the data into its two children n' and n'' (we will assume w.l.o.g. that $n' < n''$). A datum (x, y) is assigned to the "left" child n' if $y < t_n$, and

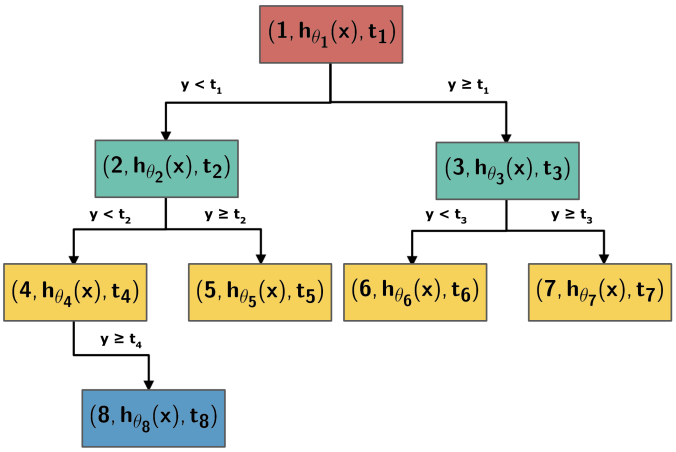


Figure 3-1: Illustration of neural regression tree. Each node is equipped with a neural classifier h_θ. The splitting threshold t depends on the target variable y and is locally optimized.

to the "right" child n'' otherwise. Then, the actual partitions of the target variable are the tree's leaves. To partition the data, each node carries a classifier $h_{\theta_n} : x \mapsto \{n', n''\}$, which assigns any instance with features x, to one of n' or n''. In our instantiation of this model, the classifier h_{θ_n} is a neural classifier that not only classifies the features but also adapts and refines the features to each node.

Given an entire tree along with all its parameters and an input x, we can compute the *a posteriori* probability of the partitions (i.e., the leaves) as follows. For any leaf l, let $l_0 \to \cdots \to l_p$ represent the chain of nodes from root l_0 to the leaf itself $l_p \equiv l$. The *a posteriori* probability of the leaf is given by $P(l \mid x) = \prod_{r=1}^{p} P(l_r \mid l_{r-1}, x)$, where each $P(l_r \mid l_{r-1}, x)$ is given by the neural classifier on node l_{r-1}. Substitution into (3.3) yields the final predicted value of the target variable

$$\hat{y}(x) = \sum_{l \in \text{leaves}} P(l \mid x) r_l(x) \tag{3.5}$$

where $r_l(x) \coloneqq r(x, l)$, in our setting, is simply the mean value of the leaf bin. Other options include the center of gravity of the leaf bin, using a specific regression function, etc.

NRT Optimization

We optimize each node of the NRT tree individually. The procedure to optimize an individual node n is as follows. Let $\mathbb{D}_n = \{(x_i, y_i)\}$ represent the set of training samples arriving at node n. Let n' and n'' be the children nodes induced by the threshold t_n. In principle, to locally optimize n, we must minimize the average regression error $\mathcal{E}(\mathbb{D}_n; t_n, \theta_n) = \text{avg}\left(\mathcal{E}(y, \hat{y}_n(x))\right)$ between the true target y and the estimated target $\hat{y}_n(x)$ computed using only the subtree rooted at n. In practice, $\mathcal{E}(\mathbb{D}_n; t_n, \theta_n)$ is not computable, since the subtree at n is as yet unknown. Instead, we will approximate it through the classification accuracy of the classifier at n, with safeguards to ensure that the resultant classification is not trivial and permits useful regression.

Let $y(t_n) = \text{sign}(y - t_n)$ be a binary indicator function that indicates if an instance (x, y) has to be assigned to child n' or n''. Let $\mathcal{E}(y(t_n), h_{\theta_n}(x))$ be a qualifier of the classification error (which can be binary cross entropy loss, hinge loss, etc.) for any sample (x, y). We define the classification loss at node n as

$$E_{\theta_n, t_n} = \frac{1}{|\mathbb{D}_n|} \sum_{(x,y) \in \mathbb{D}_n} \mathcal{E}(y(t_n), h_{\theta_n}(x)) \tag{3.6}$$

where $|\mathbb{D}_n|$ is the size of $\mathbb{D}_n$. The classification loss (3.6) cannot be directly minimized w.r.t t_n, since this can lead to trivial solutions, e.g., setting t_n to an extreme value such that all data are assigned to a single class. While such a setting would result in perfect classification, it would contribute little to the regression. To prevent such solutions, we include a triviality penalty $\mathcal{T}$ that attempts to ensure that the tree remains balanced in terms of the number of samples at each node. For our purpose, we define the triviality penalty at any node as the entropy of the distribution of samples over the partition induced by t_n (other triviality penalties such as the Gini index [15] may also apply)

$$\mathcal{T}(t_n) = -p(t_n) \log p(t_n) - (1 - p(t_n)) \log(1 - p(t_n)) \tag{3.7}$$

where

$$p(t_n) = \frac{\sum_{(x,y)\in\mathbb{D}_n}(1 + y(t_n))}{2|\mathbb{D}_n|}$$

The overall optimization objective of node n is

$$\theta_n^*, t_n^* = \arg\min_{\theta_n, t_n} \lambda E_{\theta_n, t_n} + (1 - \lambda)\mathcal{T}(t_n) \tag{3.8}$$

where $\lambda \in (0, 1)$ is used to assign the relative importance of the two components of the loss and is a hyper-parameter to be tuned.

In the optimization of (3.8), the loss function depends on t_n through $y(t_n)$, which is a discontinuous function of t_n. We have two possible ways to overcome this difficulty: the scan and the gradient method. In the first, we can scan through all possible values of t_n to select the one that results in a minimal loss. Alternatively, a faster gradient-descent approach is obtained by making the objective differentiable w.r.t. t_n. Here the discontinuous function $\text{sign}(y - t_n)$ is approximated by a differentiable relaxation: $y(t_n) = 0.5(\tanh(\beta(y - t_n)) + 1)$, where β controls the steepness of the function and must typically be set to a large value ($\beta = 10$ in our settings) for close approximation. The triviality penalty is also redefined (to be differentiable w.r.t. t_n) as the proximity to the median $\mathcal{T}(t_n) = \|t_n - \text{median}(y \mid (x, y) \in \mathbb{D}_n)\|_2$, since the median is the minimizer of (3.7). We use coordinate descent to optimize the resultant loss.

Once optimized, the data $\mathbb{D}_n$ at n is partitioned into n' and n'' w.r.t. threshold t_n^*, and the process proceeds recursively down the tree. The tree's growth may be continued until the regression performance on a held-out set saturates. Algorithm 3.1 describes the entire training algorithm.

3.2.3 Experiments

To demonstrate the utility of the proposed approach, we conduct experiments on a pair of notoriously challenging regression tasks in VFAH—estimating the age and height of speakers from their voice [1, 4, 7, 31, 32, 33, 34, 35, 36]. Our model performs

Algorithm 3.1: NRT Optimization. The tree is built recursively. For each node n, the neural classifier adapts and classifies the features and partitions the data based on the locally optimal classification threshold.

Input: data $\mathbb{D}$
Parameter: $\{t_n\}, \{\theta_n\}$
Output: $\{t_n^*\}, \{\theta_n^*\}$
Function `BuildTree`$(\mathbb{D}_n)$**:**
    Initialize t_n, θ_n
    $t_n^*, \theta_n^* \leftarrow$ `NeuralClassifier`$(\mathbb{D}_n, t_n, \theta_n)$
    $\mathbb{D}_{n'}, \mathbb{D}_{n''} \leftarrow$ `Partition`$(\mathbb{D}_n, t_n^*)$
    for $\mathbb{D}_n$ **in** $\{\mathbb{D}_{n'}, \mathbb{D}_{n''}\}$ **do**
        if $\mathbb{D}_n$ **is** pure **then**
            Continue
        else
            `BuildTree`$(\mathbb{D}_n)$
        end
    end
end
`BuildTree`$(\mathbb{D})$

significantly better than other regression models, including those known to achieve the current state-of-the-art in these problems.

Data

To promote a fair comparison, we select two well-established public datasets in the speech community. For age estimation, we use the Fisher English corpus [37]. It consists of a 2-channel conversational telephone speech for $11,971$ speakers, comprising $23,283$ recordings. After removing 58 speakers with no age specified, we are left with $11,913$ speakers with $5,100$ male and $4,813$ female speakers. To the best of our knowledge, the Fisher corpus is the largest English language database that includes the speaker's age information for the age estimation task. The data division for the age estimation task is shown in Table 3.1. The division is made through stratified sampling such that there is no speaker overlap, and all age groups are represented across the splits. Furthermore, Figure 3-2 shows the age distribution of the dataset for the three splits (train, development, test) in relation to the Table 3.1.

For height estimation, we use the NIST speaker recognition evaluation (SRE) 2008

Table 3.1: Fisher Dataset Statistics

	# of Speakers / Utterances	
	Male	Female
Train	3,100 / 28,178	4,813 / 45,041
Dev	1,000 / 9,860	1,000 / 9,587
Test	1,000 / 9,813	1,000 / 9,799

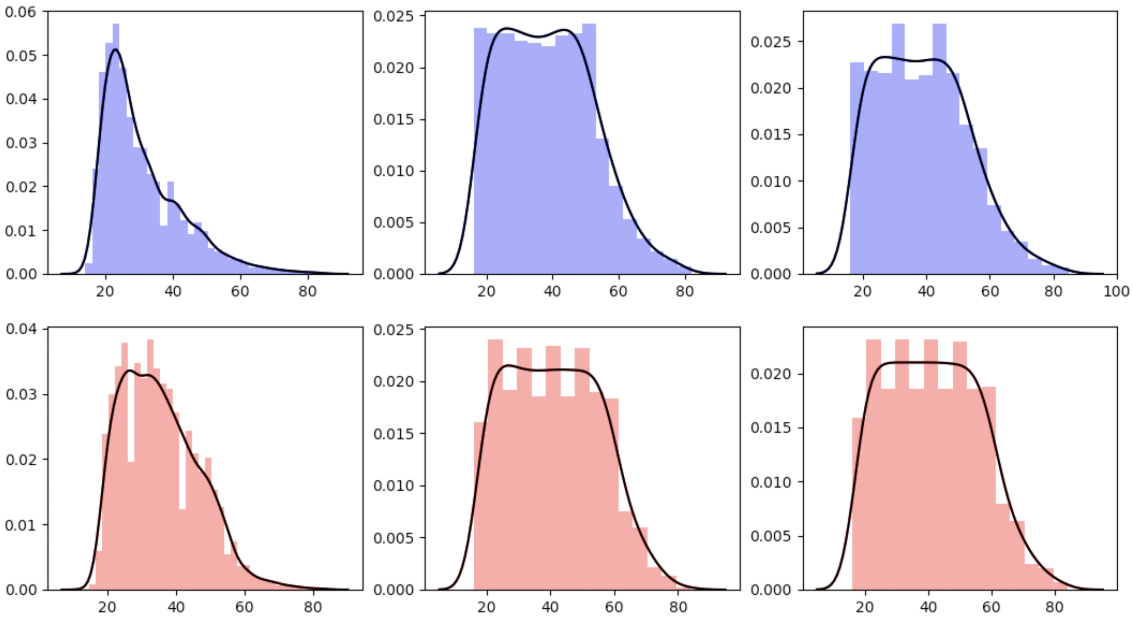


Figure 3-2: Age distribution (in percentages) for male (*Top*) and female (*Bottom*) speakers for the Fisher database for train (*Left*), development (*Center*) and test (*Right*) sets. The horizontal axis is age.

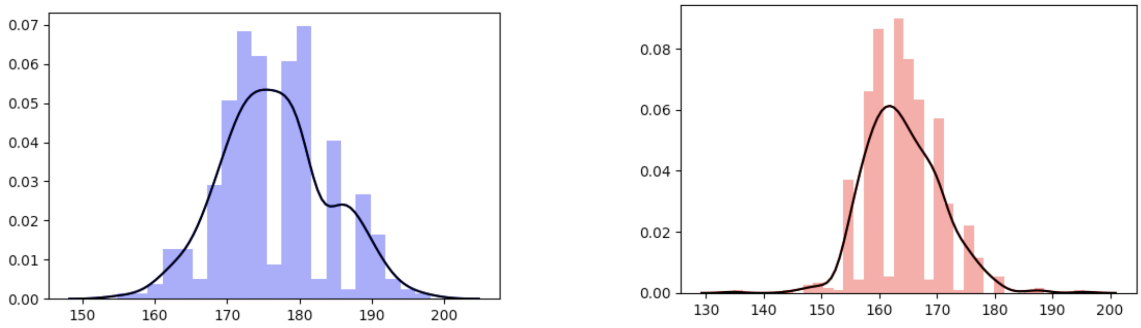
corpus [38]. We only obtain heights for 384 male and 651 female speakers. We evaluate this task using cross-validation because of data scarcity. Table 3.2 and Figure 3-3 show the statistics for the NIST-SRE8 dataset.

Since the recordings for both datasets have plenty of silences and the silences do not contribute to the information gain, Gaussian-based voice activity detection (VAD) is performed on the recordings. Then, the resulting recordings are segmented into one-minute segments.

To properly represent the speech signals, we adopt one of the most influential and well-studied representations—i-vectors [39]. I-vectors are statistical low-dimensional representations over the distributions of spectral features and are commonly used in

Table 3.2: NIST-SRE8 Dataset Statistics

# of Speakers / Utterances	
Male	Female
384 / 33,493	651 / 59,530



(a) (b)

Figure 3-3: Height distribution (in percentages) for **(a)** male and **(b)** female speakers for the NIST-SRE8 dataset. The horizontal axis is height.

state-of-the-art speaker recognition systems [40] and age estimation systems [41, 42]. Respectively, 400-dimensional and 600-dimensional i-vectors are extracted for Fisher and SRE datasets using the state-of-the-art speaker identification system [43].

Specifically, we first calculate the 60-dimensional Mel frequency cepstral coefficients (MFCCs) and then train the universal background model (UBM) on 3500 speakers with 512 Gaussian mixtures. Next, we train the 400-dimensional i-vector projector on 9346 speakers. Finally, we project all utterances to their i-vectors. The i-vector extraction process is implemented with the Bob toolbox [44, 45]. To maximize the performance of SVM-based classification, we further employ compensation strategies for the i-vectors. We apply Fisher linear discriminant analysis (LDA) with the subspace dimension of 50, within-class covariance normalization (WCCN), and length normalization (LN) sequentially [46]. The i-vectors for the NIST-SRE dataset are extracted similarly using the Kaldi SRE10 recipe [47]. The UBM has 2048 Gaussian components, and the i-vectors are 600-dimensional.

Models

We compare our model with (1) a regression baseline using the support vector regression (SVR) [48], (2) a regression tree baseline using classification and regression tree (CART) [15], and (3) a neural net baseline with multi-layer perceptron (MLP). Furthermore, to show the effectiveness of the "neural part" of our NRT model, we further compare our neural regression tree with another baseline (4) regression tree with the support vector machine (SVM-RT).

The proposed NRT is a binary tree with neural classification models, where the neural classifiers are 3-layer feedforward neural nets. Each model is associated with a set of hyper-parameters that have to be tuned on the development set, such as the λ in (3.8); the number of neurons and layers, batch size, learning rate for the neural nets; the margin penalty, kernel type and bandwidth for SVM and SVR; the depth for CART, etc. These hyperparameters control the complexity and generalization ability of the corresponding models. We tune them based on the bias-variance trade-off until the best performance on the development set has been achieved. Table 3.3 shows the specifications for our model and the baseline models.

Results

To measure the performance of our models on the age and height estimation tasks, we use the mean absolute error (MAE) and the root mean squared error (RMSE) as evaluation metrics. The results are summarized in Table 3.4. To reduce the effect of weights initialization on the performance of models consisting of neural nets, we run those models multiple (10) times with different initializations and report the average performance error.

For age and height estimation, we observe that the proposed neural regression tree model generally outperforms other baselines in both MAE and RMSE. For the height task, the neural regression tree has a slightly higher RMSE than SVR, indicating higher variance. This is reasonable as our NRT does not directly optimize the mean squared error. Bagging or forest mechanisms may be used to reduce the variance.

Table 3.3: Model Specifications

Model	**Specification**	
	Age	Height
NRT	Linear: (400, 1000) Linear: (1000, 1000) Linear: (1000, 1) Nonlin.: ReLU Optim.: Adam (lr 0.001)	(Same with Age except input dim. 600)
SVM-RT	Kernel: RBF Regul.: ℓ_1 Optim.: Scan	(Same with Age except linear kernel)
SVR	Kernel: RBF Regul.: ℓ_1	Kernel: Linear Regul.: ℓ_1
CART	Criterion: MSE	Criterion: MSE
MLP	Linear: (400, 512) Linear: (512, 1) Nonlin.: ReLU Optim.: Adam (lr 0.01)	Linear: (600, 2048) Linear: (2048, 1) Nonlin.: ReLU Optim.: Adam (lr 0.005)

Furthermore, with the neural classifier in NRT being replaced by an SVM classifier (SVM-RT), we obtain higher errors than in NRT, demonstrating the effectiveness of the neural part of the NRT as it enables the features to refine with each partition and adapt to each node. Nevertheless, SVM-RT still yields smaller MAE and RMSE values than SVR and CART and is on par with the MLP on the age task. On the height task, SVM-RT outperforms SVR, CART, and MLP in terms of MAE values with relatively small variances. This consolidates our claim that even without using a neural network, our model can find optimal thresholds for the discretization of the target variable. On the other hand, this also confirms that using neural nets without tree adaptation only contributes to a small portion of the performance gain, provided that the neural nets generalize well. Additionally, we observe that a simple-structured MLP, compared to the MLP component in NRT, is required to obtain reasonable performance—a more complex-structured MLP would not generalize well to the test set and yield high estimation bias. This, in turn, implies that our NRT model can employ high-complexity neural nets to adapt the features to be more discriminative

Table 3.4: Experiment Results

Task	Dataset	Method	**Male**		**Female**	
			MAE	RMSE	MAE	RMSE
Age	Fisher	SVR	9.22	12.03	8.75	11.35
		CART	11.73	15.22	10.75	13.97
		MLP	9.06	11.91	8.21	10.75
		SVM-RT	8.83	11.47	8.61	11.17
		NRT	**7.20**	**9.02**	**6.81**	**8.53**
Height	SRE	SVR	6.27	6.98	5.24	**5.77**
		CART	8.01	9.34	7.08	8.46
		MLP	8.17	10.92	7.46	9.47
		SVM-RT	5.70	7.07	4.85	6.22
		NRT	**5.43**	**6.40**	**4.27**	6.07

as the discretization refines and, simultaneously, maintain the model's generalization.

To test the significance of the results, we further conduct pairwise statistical significance tests. We hypothesize that the errors achieved from our NRT method are significantly smaller than the closest competitor, SVR. Paired t-tests for SVR *v.s.* SVM-RT and SVM-RT *v.s.* NRT yield p-values less than 2.2×10^{-16}, indicating significant improvement. Similar results are obtained for height experiments as well. Hence, we validate the significance of the performance improvement of our NRT method in estimating age and height over the baseline methods.

Node-Based Error Analysis

The hierarchical nature of our formulation allows us to analyze the model on every level and node of the tree in terms of its classification and regression error. Figure 3-4 shows the per-level regression errors in terms of MAE for female and male speakers, where the nodes represent the age thresholds used as splitting criteria at each level, and the edges represent the regression errors. We notice that regression error increases from left to right for female and male speakers (except for the leftmost nodes, possibly due to data scarcity issues), meaning the regression error for the younger speakers is lower than the error for older speakers. In other words, our model can discriminate better between

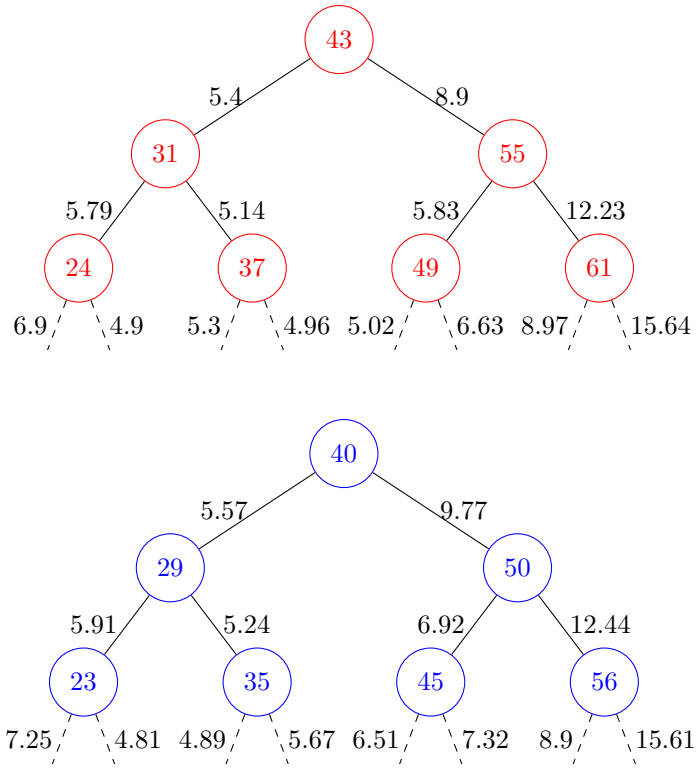


Figure 3-4: Regression errors for different age groups for female (*Top*) and male (*Bottom*) for the age estimation task. Each node represents the age threshold used as a splitting criterion, and each edge represents the regression error in terms of MAE.

younger speakers. This is in agreement with the fact that the vocal characteristics of humans undergo noticeable changes during earlier ages and then relatively stabilize for a specific age interval [49]. Hence, the inherent structural properties of our model not only improve the overall regression performance, as we see in the previous section, but also model the real-world patterns of aging in the case of age estimation.

Triviality Loss

Since our model relies on the target variable to make the partition, it is imperative to avoid any trivial partitions. A trivial partition would be where, for example, all the training samples fall into a single category. To account for that, we introduce a novel abstraction of the trivial partition—the triviality loss (3.7). Figure 3-5 shows how the triviality loss of the age estimation task increases from top to bottom for

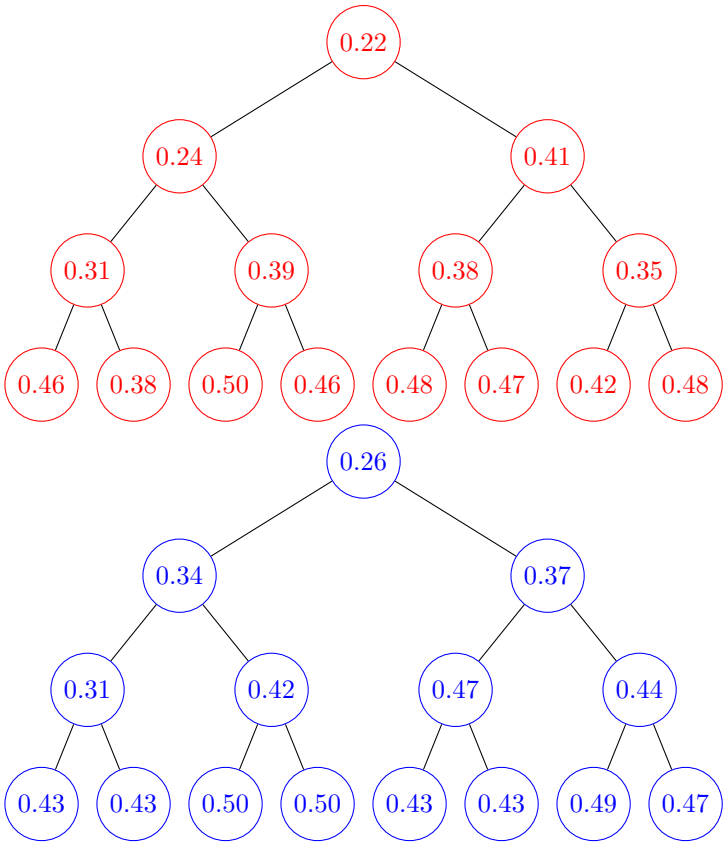


Figure 3-5: The breakdown of triviality loss on each level for female (*Top*) and male (*Bottom*) speakers for the task of age estimation.

both male and female speakers. This qualitative analysis indicates that it is relatively easier for our model to distinguish between young and old speakers than between finer age groups.

3.2.4 Related Work

Regression Trees Tree-structured models have been around for a long time. Among them, the most closely related are the regression trees. A *regression tree* is a regression function in which the partitioning is performed on features x instead of target variable y. The first regression tree algorithm was presented by [50], where they propose a greedy approach to fit a piece-wise constant function by recursively splitting the data into two subsets based on the partition of the features x. The optimal split results from minimizing the impurity, which defines the homogeneity of the split. This algorithm

set the basis for a whole line of research on classification and regression trees. Improved algorithms include CART [15], ID3 [16], m5 [17], and C4.5 [18]. Recent work combines the tree-structure and neural nets to gain the power of structure and representation learning. Such work includes the convolutional decision trees [20], neural decision trees [21, 22], adaptive neural trees [23], deep neural decision forests [24], and deep regression forests [25].

We emphasize a fundamental difference between our approach and traditional regression tree-based approaches. Instead of doing the split based on the feature space, our splitting criterion is based on the target variable, enabling the features to adapt to the partitions of the target variable.

Regression via Classification (RvC) The idea for RvC was presented by [28]. Their algorithm is based on k-means clustering to categorize numerical variables. Other conventional approaches [27, 29] for the discretization of continuous values are based on equally probable (EP) or equal width (EW) intervals. EP creates a set of intervals with an equal number of elements, and EW divides them into intervals of the same range. These approaches are ad-hoc. Instead, we propose a discretization strategy to learn the optimal thresholds by improving the neural classifiers.

Ordinal Regression Because our model is essentially a method to discretize continuous values into ordered partitions, it can be somewhat compared to ordinal regression [51]. Ordinal regression is a class of regression analysis that operates on data where the target variable is categorical but exhibits an order relation. Naïve approaches for ordinal regression often simplify the problem by ignoring the ordering information and treating the target variables as nominal categories. A slightly sophisticated method [52] decomposes the target variable into several binary ones (such as via binary, ordered partition) and estimates them using multiple models. Another relevant class of approaches uses the threshold models [52]. These approaches resemble our approach in that they assume unobserved continuous target variables underlying the ordinal values and use thresholds to discretize them. Various models (such as support

vector machines and perceptrons) are used to model the underlying target variables.

Our proposed model shares many characteristics with these approaches. However, one big difference is that the partitions are predefined by the domain problem in ordinal regression and may not be optimized for statistical inference. On the other hand, our model is based on a data-driven partition strategy where partitions are optimized for a more discriminative representation of the data hierarchy and better performance at inference time.

Limitations and Future Work We acknowledge that our model might not be ubiquitous in its utility across all regression tasks. This is, however, expected and observed to be a characteristic of target-specific models—they are indeed highly specific to the target task. We hypothesize that this model will work well with tasks that can benefit from a partition-based formulation by reducing the inherent noise in the data. We empirically show this to be true for the two example tasks above. As a natural extension of this work, we expect to test our model for other standard regression tasks in the future. Furthermore, because our model formulation inherits its properties from the regression-via-classification (RvC) framework, the objective function is optimized to reduce the classification error rather than the regression error. This limits our ability to compare our model to other regression methods directly. In the future, we intend to explore ways to minimize regression error while employing the RvC framework directly. For instance, we can develop an overall objective for the neural regression tree and study the comprehensive optimization strategy.

3.2.5 Conclusions

This study proposes a neural regression tree (NRT) for the optimal discretization of target variables in regression-via-classification (RvC) tasks. It targets the two challenges in traditional RvC approaches: finding optimal discretization thresholds and selecting the optimal set of features. We develop a hierarchical discretization strategy by recursive binary partition based on the optimality of node-wise neural classifiers. Furthermore, each partition node on the tree could locally optimize features

to be more discriminative. We propose an algorithm to optimize partition thresholds, node classifiers, and node features jointly. We also present a triviality loss to avoid trivial partitions and relaxations to enable gradient-based optimization. The proposed NRT model outperforms baselines in two challenging VFAH tasks: age and height estimation, and demonstrates significant improvements. Through these models, we demonstrate the advantages and shortcomings of target-specific modeling.

References

[1] G. Dobry et al. "Supervector dimension reduction for efficient speaker age estimation based on the acoustic speech signal". In: *IEEE Transactions on Audio, Speech, and Language Processing* 19.7 (2011), pp. 1975–1985.

[2] S. M. Mirhassani, A. Zourmand, and H.-N. Ting. "Age estimation based on children's voice: A fuzzy-based decision fusion strategy". In: *The Scientific World Journal* 2014 (2014).

[3] M. Iseli, Y.-L. Shue, and A. Alwan. "Age-and gender-dependent analysis of voice source characteristics". In: *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*. Vol. 1. IEEE. 2006, pp. I–I.

[4] M. H. Bahari, M. McLaren, D. A. van Leeuwen, et al. "Speaker age estimation using i-vectors". In: *Engineering Applications of Artificial Intelligence* 34 (2014), pp. 99–108.

[5] S. Schötz. "Acoustic analysis of adult speaker age". In: *Speaker classification I*. Springer, 2007, pp. 88–107.

[6] T. Ganchev, I. Mporas, and N. Fakotakis. "Audio features selection for automatic height estimation from speech". In: *Hellenic Conference on Artificial Intelligence*. Springer. 2010, pp. 81–90.

[7] A. H. Poorjam, M. H. Bahari, V. Vasilakakis, et al. “Height estimation from speech signals using i-vectors and least-squares support vector regression”. In: *2015 38th International Conference on Telecommunications and Signal Processing (TSP)*. IEEE. 2015, pp. 1–5.

[8] H. Arsikere et al. “Automatic estimation of the first three subglottal resonances from adults’ speech signals with application to speaker height estimation”. In: *Speech Communication* 55.1 (2013), pp. 51–70.

[9] T. Ganchev, I. Mporas, and N. Fakotakis. “Automatic height estimation from speech in real-world setup”. In: *2010 18th European Signal Processing Conference*. IEEE. 2010, pp. 800–804.

[10] A. H. Poorjam, M. H. Bahari, et al. “Multitask speaker profiling for estimating age, height, weight and smoking habits from spontaneous telephone speech signals”. In: *2014 4th International Conference on Computer and Knowledge Engineering (ICCKE)*. IEEE. 2014, pp. 7–12.

[11] S. Dusan. “Estimation of speaker’s height and vocal tract length from speech signal”. In: *Ninth European Conference on Speech Communication and Technology*. 2005.

[12] A. B. Tsybakov. “Introduction to nonparametric estimation, 2009”. In: *URL https://doi. org/10.1007/b13794. Revised and extended from the* (2004).

[13] B. Scholkopf and A. J. Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2001.

[14] H. Takeda, S. Farsiu, and P. Milanfar. “Deblurring using regularized locally adaptive kernel regression”. In: *IEEE transactions on image processing* 17.4 (2008), pp. 550–563.

[15] L. Breiman et al. *Classification and regression trees*. Monterey, CA: Wadsworth and Brooks, 1984.

[16] J. R. Quinlan. “Induction of decision trees”. In: *Machine learning* 1.1 (1986), pp. 81–106.

[17] J. R. Quinlan et al. “Learning with continuous classes”. In: *5th Australian Joint Conference on Artificial Intelligence*. Vol. 92. World Scientific. 1992, pp. 343–348.

[18] J. R. Quinlan. *C4.5: Programs for machine learning*. Elsevier, 2014.

[19] A. Liaw, M. Wiener, et al. “Classification and regression by randomForest”. In: *R news* 2.3 (2002), pp. 18–22.

[20] D. Laptev and J. M. Buhmann. “Convolutional decision trees for feature learning and segmentation”. In: *German Conference on Pattern Recognition*. Springer. 2014, pp. 95–106.

[21] H. Xiao. “NDT: Neual decision tree towards fully functioned neural graph”. In: *arXiv preprint arXiv:1712.05934* (2017).

[22] R. Balestriero. “Neural decision trees”. In: *arXiv preprint arXiv:1702.07360* (2017).

[23] R. Tanno et al. “Adaptive neural trees”. In: *arXiv preprint arXiv:1807.06699* (2018).

[24] P. Kontschieder et al. “Deep neural decision forests”. In: *2015 IEEE International Conference on Computer Vision*. IEEE. 2015, pp. 1467–1475.

[25] W. Shen et al. “Deep regression forests for age estimation”. In: *arXiv preprint arXiv:1712.07195* (2017).

[26] L. Breiman. *Classification and regression trees*. Routledge, 2017.

[27] L. Torgo and J. Gama. “Regression using classification algorithms”. In: *Intelligent Data Analysis* 1.4 (1997), pp. 275–292.

[28] S. M. Weiss and N. Indurkhya. “Rule-based machine learning methods for functional prediction”. In: *Journal of Artificial Intelligence Research* 3 (1995), pp. 383–403.

[29] L. Torgo and J. Gama. “Regression by classification”. In: *Brazilian Symposium on Artificial Intelligence*. Springer. 1996, pp. 51–60.

[30] S. A. Memon et al. “Neural regression trees”. In: *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2019, pp. 1–8.

[31] H. Kim, K. Bae, and H. Yoon. “Age and gender classification for a home-robot service”. In: *The 16th IEEE International Symposium on Robot and Human Interactive Communication*. IEEE. 2007, pp. 122–126.

[32] F. Metze et al. “Comparison of four approaches to age and gender recognition for telephone applications”. In: *IEEE International Conference on Acoustics, Speech and Signal Processing*. Vol. 4. IEEE. 2007, pp. IV–1089.

[33] M. Li, C. Jung, and K. J. Han. “Combining five acoustic level modeling methods for automatic speaker age and gender recognition”. In: *Eleventh Annual Conference of the International Speech Communication Association*. 2010.

[34] M. Lee and K. Kwak. “Performance comparison of gender and age group recognition for human-robot interaction”. In: *International Journal of Advanced Computer Science and Applications* 3.12 (2012).

[35] B. D. Barkana and J. Zhou. “A new pitch-range based feature set for a speaker’s age and gender classification”. In: *Applied Acoustics* 98 (2015), pp. 52–61.

[36] Y. Fu and T. S. Huang. “Human age estimation with regression on discriminative aging manifold”. In: *IEEE Transactions on Multimedia* 10.4 (2008), pp. 578–584.

[37] C. Cieri, D. Miller, and K. Walker. “The Fisher corpus: a resource for the next generations of speech-to-text.” In: *LREC*. Vol. 4. 2004, pp. 69–71.

[38] S. S. Kajarekar et al. “The SRI NIST 2008 speaker recognition evaluation system”. In: *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE. 2009, pp. 4205–4208.

[39] N. Dehak et al. “Front-end factor analysis for speaker verification”. In: *IEEE Transactions on Audio, Speech and Language Processing* 19.4 (2011), pp. 788–798.

[40] S. O. Sadjadi, S. Ganapathy, and J. W. Pelecanos. "Speaker age estimation on conversational telephone speech using senone posterior based i-vectors". In: *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE. 2016, pp. 5040–5044.

[41] P. G. Shivakumar et al. "Simplified and supervised i-vector modeling for speaker age regression". In: *IEEE International Conference on Acoustics, Speech and Signal Processing*. 2014, pp. 4833–4837.

[42] J. Grzybowska and S. Kacprzak. "Speaker age classification and regression using i-vectors." In: *INTERSPEECH*. 2016, pp. 1402–1406.

[43] H. Dhamyal et al. "Optimizing neural network embeddings using pair-wise loss for text-independent speaker matching". In: *INTERSPEECH*. 2018.

[44] A. Anjos et al. "Bob: a free signal processing and machine learning toolbox for researchers". In: *20th ACM Conference on Multimedia Systems*. 2012.

[45] A. Anjos et al. "Continuously reproducing toolchains in pattern recognition and machine learning experiments". In: *International Conference on Machine Learning*. 2017.

[46] J. H. Hansen and T. Hasan. "Speaker recognition by machines and humans: A tutorial review". In: *IEEE Signal processing magazine* 32.6 (2015), pp. 74–99.

[47] D. Povey et al. "The Kaldi speech recognition toolkit". In: *IEEE 2011 workshop on automatic speech recognition and understanding*. CONF. IEEE Signal Processing Society. 2011.

[48] D. Basak, S. Pal, and D. C. Patranabis. "Support vector regression". In: *Neural Information Processing-Letters and Reviews* 11.10 (2007), pp. 203–224.

[49] E. T. Stathopoulos, J. E. Huber, and J. E. Sussman. "Changes in acoustic characteristics of the voice across the life span: measures from individuals 4–93 years of age". In: *Journal of Speech, Language, and Hearing Research* 54.4 (2011), pp. 1011–1021.

[50] J. N. Morgan and J. A. Sonquist. “Problems in the analysis of survey data, and a proposal”. In: *Journal of the American Statistical Association* 58.302 (1963), pp. 415–434.

[51] P. McCullagh. “Regression models for ordinal data”. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 42.2 (1980), pp. 109–127.

[52] P. A. Gutiérrez et al. “Ordinal regression methods: survey and experimental study”. In: *IEEE Transactions on Knowledge and Data Engineering* 28.1 (2016), pp. 127–146.

Chapter 4

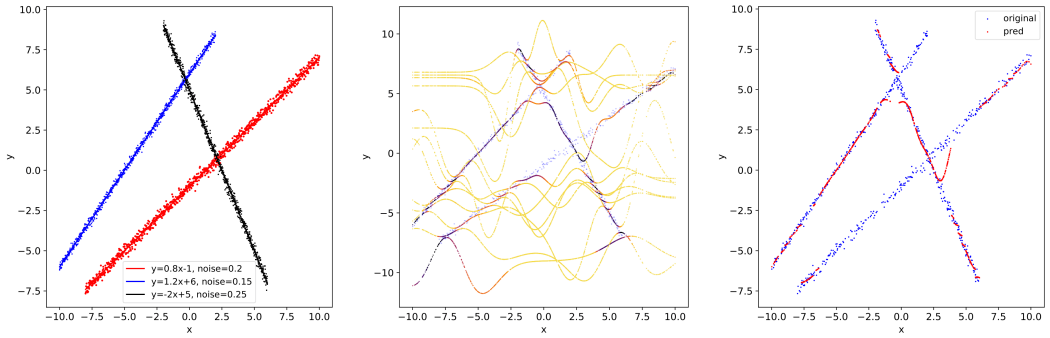
Target-Specific Models for Complicated Distributions

The last chapter presents the neural regression tree (NRT) model for the task of age and height estimation. This chapter improves the NRT model for complicated data distributions and proposes a new modeling approach: the hierarchical routing mixture of experts (HRME).

4.1 Introduction

One of the challenges in modeling a regression task is dealing with data that have complicated distributions. The distribution can be multimodal, rendering any single regression model highly biased. For instance, Figure 4-1a shows a synthetic data set uniformly sampled from three intersecting lines with different amounts of noise. A single regression model would fail to capture the multi-modality of this data and yield poor performance. This necessitates another strategy through divide-and-conquer to partition the input space into simple sub-regions and assign a regression model to each sub-region.

Many partition-based models employ this strategy. For example, decision trees [1] and random forests [2] divide the input space by hard-partitioning of feature dimensions and make piece-wise linear predictions on each partition. Mixture models [3] and



(a) (b) (c)

Figure 4-1: **(a)** A toy example: synthetic 3-lines data with different amounts of noises. **(b)** Predictions made by experts in our HRME model. Each curve represents the prediction made by one expert. Darker color indicates stronger prediction confidence. **(c)** Prediction made by our HRME model via selecting the top-1 experts.

mixtures-of-experts [4] perform soft-partition on the input space and assign regression models to each of the partitions. In particular, the mixture of expert models is tree-structured with a gating mechanism to partition the input space and a collection of experts at the leaves to make local predictions.

Although well-studied and proven effective, these models do not leverage the input-output dependency of the data distributions. For instance, as in our toy example, different regions of the output space (the y label) correspond to different data modes. Solely partitioning the input space would fit multiple data modes into each partition, requiring complex regression models to capture each input-output relation.

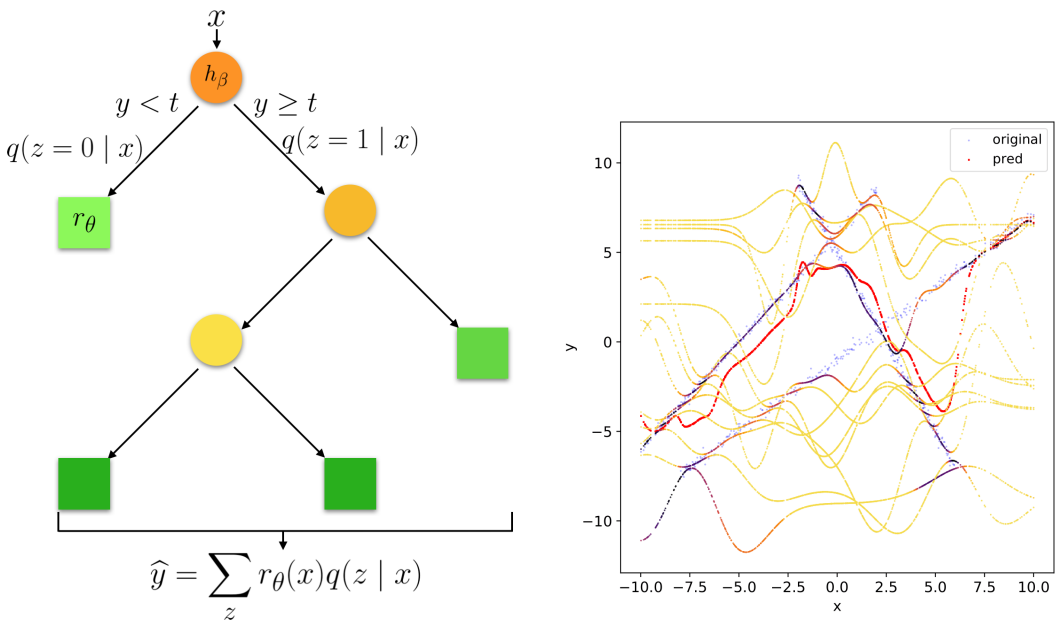
To address this issue, the neural regression tree (NRT) [5] was proposed (in the previous chapter) to partition the output space—it uses hierarchical regression-via-classification (RvC) to divide the data through an optimal output space partition and, at the same time, to optimize local features. However, it has a few disadvantages. First, it is a greedy approach and is locally optimal. Second, it does not leverage the input-output dependency of multimodally-distributed data, and strong local models such as neural nets are required to make reliable predictions.

The issues mentioned above of conventional partition-based regression methods and the NRT model can be resolved by partitioning the input-output space such that each partition only requires a simple local regression model. To accomplish this, we

propose a hierarchical routing mixture of experts (HRME) model [6], which separates the modes in the multimodal data distribution by jointly soft-partitioning the input and output spaces and makes probabilistic inferences by assigning simple regression models to each of the resultant partitions.

HRME is binary-tree structured and has two types of experts—margin-based classifiers as non-leaf node experts and simple regression models as leaf node experts. The *non-leaf node experts* function as a new gating mechanism to soft-partition the data based on their modes, predicated on the distribution of input and output variables. A node-specific binary classifier performs the partitioning. Together, the non-leaf node classifiers hierarchically partition the space into many regions. Each region corresponds to a leaf node in the tree, and within the region, the relationship between input and output variables is ideally unimodal. The *leaf node experts* make predictions on each resulting partition. These leaf node experts can now be relatively simple if the data is well partitioned. The basic assumption is that the multi-modally distributed data are nevertheless locally (and non-linearly) separable, and hence the non-leaf experts of the tree function as a routing mechanism to partition the data into subsets of simple (uni-modal) distributions and route each subset to a simple leaf expert to make predictions.

However, the actual distributions of the data and its modes are unknown *a priori*. Consequently, the binary classes for each classifier (non-leaf) node are unknown. This effectively makes the partition of the output space itself a variable to be determined. To address this, we develop a probabilistic framework for our HRME model and propose a recursive Expectation-Maximization (EM) based algorithm to optimize the joint input-output partition and the expert models. Notably, the tree structure is also optimized such that no extra pruning is required. Compared to NRT, the HRME model is globally rather than locally optimized. Figure 4-2a shows the probabilistic structure of the HRME model. Figure 4-2b shows the predictions made by the experts in the HRME model to fit the toy example in Figure 4-1a. The three intersecting lines in the synthetic example have different noise levels, which is difficult to fit for conventional regression models.



(a) (b)

Figure 4-2: **(a)** Illustration of the HRME model. It is a probabilistic binary tree. Each non-leaf node (circle) carries a classifier h_β and a partition threshold t, and each leaf node (square) carries a regressor r_θ. Prediction is made via a probabilistic combination of leaf regressors. The model is learned via recursive EM. **(b)** Predictions made by HRME experts to fit a three-line example. Each curve represents an expert prediction, with darker color indicating higher confidence. The red curve is the prediction made by combining all experts.

4.1.1 Related Work

Decision trees [1, 7, 8] are a family of supervised learning methods that utilize a partition on the input feature space and make piece-wise linear predictions. Based on them, random forests [2, 9] take an ensemble learning approach by aggregating a collection of decision trees to reduce the over-fitting tendency of a single decision tree. An issue with these tree-based methods is that they rely on hard partitions and piece-wise linear predictions, leading to discontinuities and high biases in predictions.

On the other hand, our HRME model can be viewed as a new member of the mixture models [3] and mixtures-of-experts [4]. The mixture of experts (ME) is a probabilistic tree-structured model with a gating mechanism and a collection of experts at the leaves. The gating mechanism is responsible for soft partitioning the input space into sub-regions such that a local expert models the distribution of each sub-region [10].

The flexibility of the ME family embraces a wide variety of gating mechanisms and expert models. Examples include hierarchical mixture of experts (HME) [11] which employs a binary tree structure, Bayesian HME [12] with a Bayesian approach, mixture of Gaussian processes (HME-GP) [13, 14, 15, 16], mixture of support vector machines (HME-SVM) [17, 18], etc, to name only a few.

However, the ME models have three issues:

1. The gating mechanism does not explicitly leverage the input-output dependencies of the data. Rather, it performs probabilistic input-space partitioning, based on assumed data distributions such as the multinomial distribution [11], Gaussian distribution [15], Dirichlet process [14], Gaussian process [13], etc.

2. In ME models, strong experts are often needed to achieve good performance [10].

3. The structure of the ME models, namely the tree depth and the number of experts, is often optimized through additional procedures, such as pruning [19] and Bayesian model selection [12, 20]. This increases the complexity of model learning.

Our HRME model addresses the issues with these conventional methods by (1) joint soft-partitioning of the input-output space based on the natural separability of the multimodal data and (2) joint optimization of the tree structure and the expert models without extra pruning procedures.

4.2 Hierarchical Routing Mixture of Experts

This section presents the HRME model's specifications, formulates the optimization objective, and develops the optimization algorithm.

4.2.1 Model Specification

Figure 4-2a shows the structure of the tree model. It is a binary tree. In this case, each non-leaf node is equipped with a classification expert, a binary classifier. The

node classifiers hierarchically refine the data into separate modes until the data is uni-modal at leaf nodes. Each leaf node is equipped with a regression expert, a simple linear model.

We denote the input features as $\boldsymbol{x} \in \mathbb{R}^d$ and the continuous output label as $y \in \mathbb{R}$. Separating the data modes requires determining the optimal classifier at each non-leaf node. However, we do not have the data class information beforehand, i.e., we do not know how data can be locally separated. As a remedy, we adopt a thresholding strategy—setting a threshold t on y such that $y = 0$ if $y < t$ and $y = 1$ otherwise. As a result, we assign binary classes to data via thresholding on y. However, note that by doing so, we effectively make t a variable to be optimized; namely, we are not only partitioning on $\boldsymbol{x}$ but also partitioning on y. We will explain the optimization of this joint partition in a later section.

At this point, let us assume we have known the optimal tree settings—that is, we know the tree structure (the depth and the number of nodes), and for each non-leaf node, the optimal splitting threshold t^* and the classifier h_{β^*} parameterized by the optimal parameter β^*, and for each leaf node, the regressor r_{θ^*} parameterized by the optimal parameter θ^*. We then explain the prediction of y given an input $\boldsymbol{x}$.

Specifically, for notational convenience, we assume that the nodes are numbered such that for any two nodes n_i and n_j if $i < j$, n_i occurs either to the left of n_j or above it in the tree. Each node n_i carries a classifier $h_{\beta^*_{n_i}} : \boldsymbol{x} \mapsto \{n_{i+1}, n_{i+2}\}$, which assigns any instance with input $\boldsymbol{x}$ to one of the children nodes n_{i+1} or n_{i+2}. We introduce a binary-valued random variable $\mathrm{z}_{n_i} \in \{0, 1\}$ to indicate $\boldsymbol{x}$ being assigned to n_i or not. Then, the corresponding likelihood of $\boldsymbol{x}$ being assigned to node n_i is estimated by the classifier on node n_{i-1}

$$q(\mathrm{z}_{n_i} \mid \boldsymbol{x}) \equiv q(\mathrm{z}_{n_i} = 1 \mid \boldsymbol{x}) \leftarrow h_{\beta^*_{n_{i-1}}}(\boldsymbol{x}) \tag{4.1}$$

Next, we would like to know the likelihood of a data point $\boldsymbol{x}$ being routed to a specific leaf. Denote the chain from root $l_1 \equiv n_0$ to leaf l_k as $l_1 \to \ldots \to l_k$, then the likelihood

of $\boldsymbol{x}$ being assigned to leaf l_k is

$$q(\mathrm{z}_{l_k} \mid \boldsymbol{x}) = \sum_{\mathrm{z}_{l_1}} \cdots \sum_{\mathrm{z}_{l_{k-1}}} q(\mathrm{z}_{l_1}, \ldots, \mathrm{z}_{l_k} \mid \boldsymbol{x}) \tag{4.2}$$

Applying the sum-product rule and using the conditional dependency to (4.2) yields

$$q(\mathrm{z}_{l_k} \mid \boldsymbol{x}) = \prod_{j=1}^{k-1} q(\mathrm{z}_{l_{j+1}} \mid \mathrm{z}_{l_j}, \boldsymbol{x}) \tag{4.3}$$

For leaf l_k, it carries a regressor $r_{\theta^*_{l_k}}$ such that the prediction $\widehat{y}_{l_k} = r_{\theta^*_{l_k}}(\boldsymbol{x})$. Then, an estimate of y is given by the expectation of the predictions over all leaves

$$\widehat{y} = \sum_{l_k \in \text{leaves}} r_{\theta^*_{l_k}}(\boldsymbol{x}) q(\mathrm{z}_{l_k} \mid \boldsymbol{x}) \tag{4.4}$$

and the corresponding conditional density for leaf l_k is

$$p(y \mid \mathrm{z}_{l_k}, \boldsymbol{x}) \leftarrow r_{\theta^*_{l_k}}(\boldsymbol{x}) \tag{4.5}$$

4.2.2 Learning Algorithm

From the previous section, we have shown that in order to make predictions using the tree, we need to determine the optimal tree settings, i.e., the tree structure $\{n_i\}$, the non-leaf node thresholds $\{t_{n_i}\}$, the classifier parameters $\{\beta_{n_i}\}$, and the leaf node regressor parameters $\{\theta_{n_i}\}$.

We adopt a maximum-likelihood approach. Specifically, our objective is to maximize

the log-likelihood for each $\boldsymbol{x}$

$$\begin{aligned}
&\max \log p(y \mid \boldsymbol{x}) && (4.6)\\
&= \log p(y \mid \boldsymbol{x}) \frac{q(\mathrm{z} \mid \boldsymbol{x})}{q(\mathrm{z} \mid \boldsymbol{x})} \sum_{\mathrm{z}} q(\mathrm{z} \mid \boldsymbol{x}) \\
&= \sum_{\mathrm{z}} q(\mathrm{z} \mid \boldsymbol{x}) \log \frac{p(y, \mathrm{z} \mid \boldsymbol{x})}{p(\mathrm{z} \mid y, \boldsymbol{x})} \frac{q(\mathrm{z} \mid \boldsymbol{x})}{q(\mathrm{z} \mid \boldsymbol{x})} \\
&= \sum_{\mathrm{z}} q(\mathrm{z} \mid \boldsymbol{x}) \log \frac{p(y, \mathrm{z} \mid \boldsymbol{x})}{q(\mathrm{z} \mid \boldsymbol{x})} + && (4.7)\\
&\quad \sum_{\mathrm{z}} q(\mathrm{z} \mid \boldsymbol{x}) \log \frac{q(\mathrm{z} \mid \boldsymbol{x})}{p(\mathrm{z} \mid y, \boldsymbol{x})} && (4.8)
\end{aligned}$$

where $q(\mathrm{z} \mid \boldsymbol{x})$ is an estimate for the true assignment mass $p(\mathrm{z} \mid \boldsymbol{x})$; (4.7) is commonly referred to as the evidence lower bound (ELBO) which needs to be improved to maximize the log-likelihood (4.6); (4.8) is the Kullback-Leibler divergence which measures the distance of two probability masses, and is always greater than or equal to zero.

Therefore, it is natural to apply the expectation-maximization (EM) method to optimize (4.6). Specifically, in the E-step, we compute the ELBO (4.7) for all the training instances

$$\begin{aligned}
Q(p, q) &= \sum_{\boldsymbol{x}} \sum_{\mathrm{z}} q(\mathrm{z} \mid \boldsymbol{x}) \log \frac{p(y, \mathrm{z} \mid \boldsymbol{x})}{q(\mathrm{z} \mid \boldsymbol{x})} \\
&= \sum_{\boldsymbol{x}} \sum_{\mathrm{z}} q(\mathrm{z} \mid \boldsymbol{x}) \log \frac{p(y \mid \mathrm{z}, \boldsymbol{x}) p(\mathrm{z} \mid \boldsymbol{x})}{q(\mathrm{z} \mid \boldsymbol{x})} && (4.9)
\end{aligned}$$

where $q(\mathrm{z} \mid \boldsymbol{x})$ is given by (4.3), and $p(y \mid \mathrm{z}, \boldsymbol{x})$ is given by (4.5) (for example, the leaf node gives a Gaussian distribution over y if we assume a linear model with Gaussian noise). The true leaf node assignment mass $p(\mathrm{z} \mid \boldsymbol{x})$ is yet unknown. However, we can estimate it using the empirical frequency of the number of samples at the leaf node over the total number of training samples. This is a crude estimation, but we will provide a better strategy in the latter part of this section.

In the M-step, we optimize the parameters to increase the ELBO (4.7). Specifically, we optimize the non-leaf node expert to maximize the classification accuracy and the

leaf-node expert to minimize the regression error.

However, the data classes are unavailable, and the non-leaf node threshold t is unknown. We provide an alternative approach to mitigate this difficulty. For each non-leaf node, we perform a grid search over the possible values of t, and for each t, we perform the M-step. The best t value is obtained as the one with a maximum Q-value. Although different sampling strategies can be used when searching for t, grid-search works well in practice.

As we mentioned earlier, it is difficult to estimate the true leaf node assignment mass $p(\mathrm{z} \mid \boldsymbol{x})$. Although variational approximation may be used, we propose an empirically simpler strategy. Instead of using the Q-value as a global indicator of the optimality of the tree, we propose to use an alternative proxy: the negative mean-square-error

$$Q_{\text{alternative}} = -\text{mean}(y - \widehat{y})^2 \tag{4.10}$$

where $\widehat{y}$ is given by (4.4).

The recursive EM algorithm is summarized in Algorithm 4.1. We start from the root node and grow the tree recursively in a depth-first manner, i.e., from top to bottom, from left to the right. Each time we grow a three-node subtree. We keep increasing the number of nodes until the lower bound Q stops increasing or the ratio of the number of samples at the leaf to the total number of samples is below some preset threshold.

4.3 Experiments

This section evaluates our HRME model and the recursive EM algorithm on a collection of regression tasks. We describe the experimental settings and present the results for our method and a wide range of baseline methods.

Algorithm 4.1: Recursive EM for HRME

Input: $\{data\}, \{root\}$
Parameter: $\{t\}$, classifier parameters, regressor parameters
Output: HRME Tree
Function `GrowTree`$(\{data\}, \{nodes\})$**:**
  for n **in** $\{nodes\}$ **do**
    $n_l, n_r \leftarrow$ `GrowSubtree`(n)
    $\mathbb{D} \leftarrow \{data\}$
    for t **do**
      $\mathbb{D}_l, \mathbb{D}_r \leftarrow$ `SplitData`$(\mathbb{D}, t)$
      if $\frac{\min(|\mathbb{D}_l|, |\mathbb{D}_r|)}{\#\text{total samples}} <$ threshold **then**
        Continue
      end
      n.`TrainClassifier`$(\mathbb{D}, t)$
      `PropagateConditional()` using Equation (4.3)
      n_l.`TrainLeaf`$(\mathbb{D}_l)$
      n_r.`TrainLeaf`$(\mathbb{D}_r)$
      $Q \leftarrow$ `ComputeQ()` using Equation (4.10)
    end
    if $Q > Q^*$ **then**
      $Q^* \leftarrow Q$
      $\{data\} \leftarrow \mathbb{D}_l, \mathbb{D}_r$
      $\{nodes\} \leftarrow n_l, n_r$
      `GrowTree`$(\{data\}, \{nodes\})$
    else
      Remove the subtree n_l, n_r
      Continue
    end
  end
end

4.3.1 Data

For demonstration purposes, we synthesize a 3-lines dataset (as shown in Figure 4-1a). We select five other standard datasets commonly used in regression tasks for further evaluation. Four of these datasets are from the UCI machine learning repository [21]: the `CCPP` dataset [22, 23], the `concrete` dataset [24], the `Boston housing` dataset [25], and the `energy` dataset [26]. The fifth is the `kin40k` dataset [27, 28]. The datasets range from small-sized to large-sized and from low-dimensional to high-dimensional. The statistics are shown in Table 4.1. The train and test divisions either use the

Table 4.1: Dataset Statistics

Dataset	Feature Dim	Train	Test
3-lines	1	1750	750
Housing	13	354	152
Concrete	8	721	309
CCPP	4	6697	2871
Energy	28	14803	4932
Kin40k	8	10000	30000

default split or the 0.7 : 0.3 split. Further, to demonstrate the effectiveness of the HRME model on more challenging regression tasks, we test it on two VFAH tasks: age and height estimation from speech. The task and data details are described in Section 3.2.3. The same i-vector representations are used as input features to the models.

4.3.2 Models

Baselines To conduct a fair evaluation, we compare our method with a wide range of baselines: linear regression (LR), support vector regression (SVR), decision trees (DT), random forests (RF), the hierarchical mixture of experts (HME) with strong Gaussian or Gaussian process experts, and multi-layer perceptron (MLP). Each model carries a set of parameters to be estimated and hyperparameters (e.g., margin and kernels in SVR, depth and number of nodes in DT and RF, number of neurons and learning rate in MLP, etc.) to be tuned. We train the models on training sets and fine-tune the hyperparameters using grid-search and three-fold cross-validation on the training sets to obtain the best performance. The models are implemented with scikit-learn toolkit [29] or PyTorch [30]. For HME models, we obtain the best results from the literature under the same experimental settings.

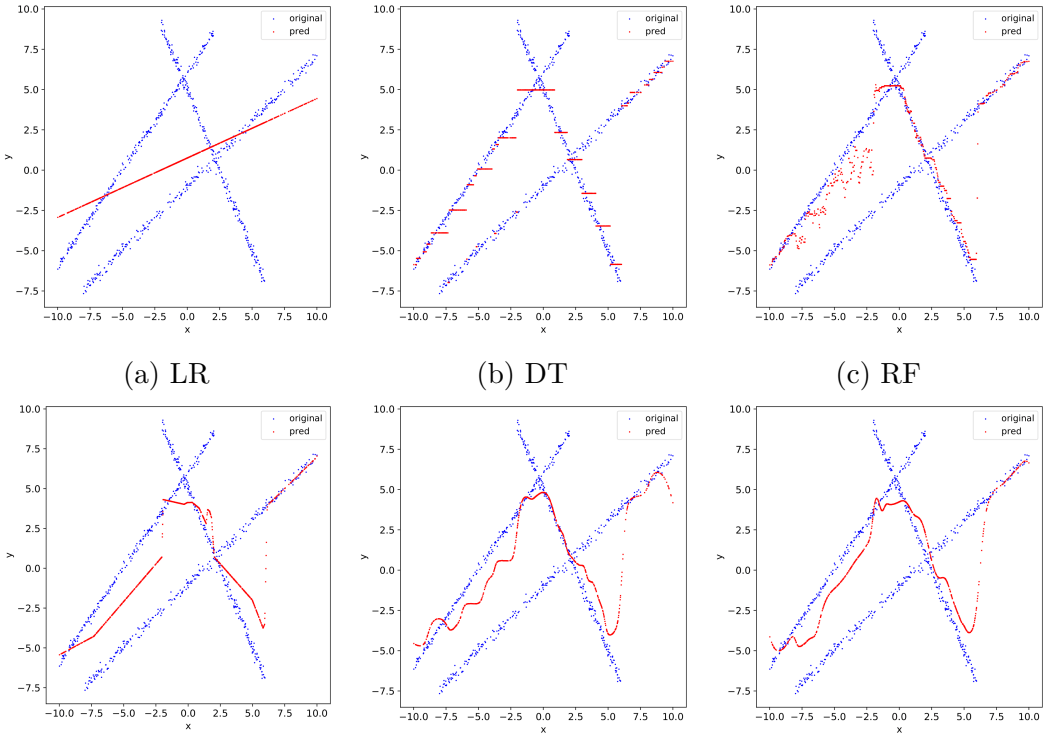
HRME Models For our HRME model, we train it following Algorithm 4.1. In our model instantiation, the non-leaf experts are support vector machines with radial basis function kernels (SVM-RBF). We choose two simple models for leaf experts,

the linear regression model (HRME-LR) or the support vector regression model with a radial basis function kernel (HRME-SVR). Similar to baselines, our models are also trained and fine-tuned on the same training sets, following the same strategy as the baseline methods. In addition, all non-leaf experts on the tree share the same hyperparameters, and so do the leaf experts. Although it would be desirable to use different hyperparameters for nodes on different tree levels as the data size shrinks with the tree depth, our model is robust to such variations. To account for the high feature dimensionality for age and height prediction tasks, we further construct an HRME-MLP variant that uses a simple two-layer multi-layer perceptron (MLP) as the non-leaf expert and SVR as the leaf expert.

4.3.3 Results

We evaluate our method and baseline methods with two metrics: the mean absolute error (MAE) and the root mean squared error (RMSE). On the synthetic 3-line data, Figure 4-3 shows the fitting results on the test set for our method and baseline methods. Our HRME models provide more accurate predictions than the baselines. Specifically, the linear model predicts the mean of the three different distributions; the decision tree and random forest provide a better fit than linear regression, but discontinuities and higher variance occur due to these two models' piece-wise linear nature. MLP achieves a smaller prediction error than DT and RF but shows discontinuities and failure to capture the data modality. In comparison, our HRME models provide much smoother fitting with lower bias and variance than the baselines. Note that even with linear leaf experts, the HRME-LR model can capture the data's nonlinear modality and make regional predictions by soft-switching its experts among the three distributions. Further, the HRME-SVR model yields smoother predictions using nonlinear leaf experts than the HRME-LR model, with lower bias and variance. Additionally, we observe that all models here prefer the upper line to the lower line due to the higher noise level in the lower line.

Figure 4-1b shows the predictions made by the experts in the HRME-SVR model. Fourteen experts (indicated by colored curves) are allocated to different data regions.



(a) LR (b) DT (c) RF

(d) MLP (e) HRME-LR (f) HRME-SVR

Figure 4-3: Fitting results on synthetic data with different models: linear regression (LR), decision tree (DT), random forest (RF), multi-layer perceptron (MLP), our HRME with linear regressor (HRME-LR) and SVR regressor (HRME-SVR).

Each expert is confident of making predictions within one data mode, as indicated by higher posterior probabilities (darker colors), and all data modes are successfully captured. Consequently, if we have prior knowledge of the data distribution, this could be used to select the experts for making the best predictions. Further, instead of using the weighted average of all experts, we select the top-1 expert to make predictions. Figure 4-1c shows the corresponding fitting results. We see a much better fit than in Figure 4-3f—in the former, our HRME-SVR model successfully predicts all data modes.

We further show the growth of the HRME tree on the training set. In Figure 4-4, the number in each circled node is the partition threshold t. The number beside each circle is the RMSE if growth stops at that node. The tree is grown depth-first (top to bottom, left to right). We observe that the RMSE reduces as the tree grows. This validates our hypothesis that our algorithm can automatically learn the optimal tree

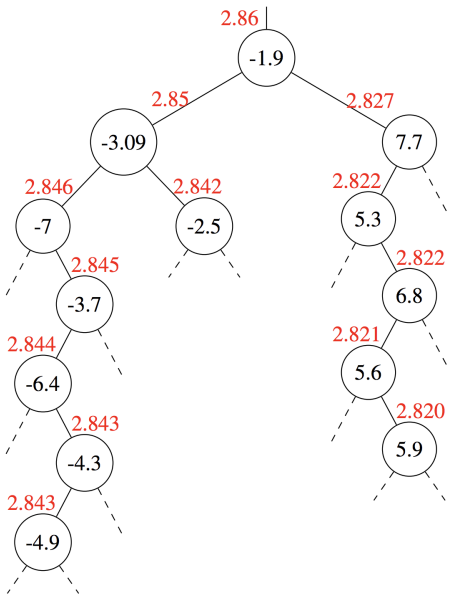


Figure 4-4: The HRME tree after training on the synthetic data. The tree is grown recursively in a depth-first manner—top to bottom, left to right. Each circle represents a classifier node, and the number within it is the partition threshold t. The number on edge represents the root mean square error if it stops growing at that node. Each dashed edge leads to a leaf regressor.

structure without pruning afterward, and the proposed Q-value is a good indicator of the tree's global optimality. Further, our HRME model also successfully partitions the output space based on the separability of data modes by finding the thresholds like -1.9, 5.6, -6.4, etc.

Table 4.2 shows comprehensive results for all the methods on all the datasets. We observe an overall improvement of our HRME methods over the baseline methods. Specifically, for large datasets like `engery` and `kin40k`, our methods outperform all other baselines in terms of bias (MAE) and variance (RMSE), even for the HME models with strong Gaussian process experts and the MLP. For medium-sized datasets like `CCPP` and `concrete`, our methods generally outperform other baselines except RF. Nevertheless, like RF, our method can also be ensembled or boosted (now averaged) to improve performance [31]. For small datasets like `Boston housing`, our methods do not outperform DT and RF. However, on a closer look, we find that HRME-LR yields much smaller MAE and RMSE than HRME-SVR and is on par with DT and

RF. This observation indicates the linear nature of the data distribution; hence, a nonlinear regression expert would be inappropriate for this dataset. This observation is also confirmed by the poor performance of the nonlinear MLP model. Further, the data is small (506 samples) but has a high dimensionality (13), making it difficult to separate the modes by SVM. Instead, other classifiers can be used to improve the performance of our model. We also observe that our methods can reduce the variance (RMSE) on most tasks. This shows that our methods can mitigate the high variance problem of conventional tree models. In addition, we see that even with simple linear leaf experts, our method can significantly outperform LR and compete with nonlinear models like SVR, RF, and MLP. This validates our hypothesis that simple leaf experts can make good predictions with our data modality-aware routing mechanism. Lastly, MLP performs poorly in most tasks, even with fine-tuning. This shows that MLP cannot capture the specific modality of data distributions in this case.

Table 4.3 shows the results for age and height estimation from speech. The SVM-RT and NRT results are taken from Section 3.2.3 in Chapter 3. HRME achieves better performance than NRT in terms of estimation error and variance—HRME with SVM experts (HRME-SVR) performs better than its SVM-RT counterpart, and HRME with MLP experts (HRME-MLP) performs better than its NRT counterpart, suggesting that HRME enjoys global optimality. In contrast, SVM-RT and NRT only achieve local optimality. In both comparisons, HRME yields smaller variances, implying robustness to overfitting. HRME-MLP performs better than HRME-SVR, indicating MLP non-leaf experts can better classify high-dimensional features than SVM experts. Further, compared to NRT, which uses 3-layer MLP node classifiers, HRME-MLP uses simpler 2-layer MLP node experts, confirming our claim that HRME can optimally partition the complexly-distributed data into simple modes such that simple experts can make reliable predictions.

To this point, comprehensive experiment results show that our HRME methods perform well on various regression tasks, especially on large, high-dimensional, and complicated datasets. Our HRME methods can capture the complex data hierarchy, reduce variance, and make good predictions with simple leaf experts. We further

Table 4.2: Standard Regression Task Results

Dataset	Metric	LR	SVR	DT	RF	HME	MLP	HRME	
								LR	SVR
3-lines	MAE	3.352	2.006	2.224	2.131	—	**1.960**	2.337	2.250
	RMSE	4.104	3.173	3.291	3.072	—	**2.795**	2.885	2.859
Housing	MAE	3.651	3.498	2.537	**2.103**	4.170[1]	6.711	2.682	3.266
	RMSE	4.911	5.126	3.665	**3.043**	5.610[2]	8.535	3.857	4.376
Concrete	MAE	8.088	8.013	4.919	**3.436**	—	5.394	4.121	4.020
	RMSE	10.204	10.772	8.000	**4.806**	6.250[3]	6.594	5.664	5.609
CCPP	MAE	3.601	2.746	2.941	**2.383**	—	4.013	2.965	2.712
	RMSE	4.578	3.856	4.151	**3.409**	4.100[4]	5.078	3.951	3.805
Energy	MAE	52.075	43.141	43.996	52.002	—	40.521	42.121	**40.009**
	RMSE	93.564	101.267	99.654	95.558	—	88.191	89.203	**87.022**
Kin40k	MAE	0.806	0.092	0.592	0.433	—	0.237	0.150	**0.071**
	RMSE	0.996	0.161	0.773	0.548	0.230[5]	0.312	0.212	**0.114**

[1] using Gaussian experts; results taken from [32]
[2] using Gaussian experts; results taken from [32]
[3] using Gaussian process experts; results taken from [33]
[4] using Gaussian process experts; results taken from [33]
[5] using Gaussian process experts; results taken from [16]

explore some theoretical properties of our HRME model.

4.4 Convergence and Complexity Analysis

Convergence Let n, d, and k be the number of training samples, the dimension of each sample, and the number of experts, respectively. In [34], authors prove that with large samples, the ME models can uniformly approximate Sobolev class functions of order r in the L_p norm at a rate of at least $\mathcal{O}(Ck^{-r/d})$ with constant C. This bounds the approximation error of the general ME family. Further, in [35], the authors prove that the HME mean functions can approximate the true mean function at a rate of $\mathcal{O}(k^{-2/d})$ in the L_p norm. Authors in [36] also show that the HME probability density functions can approximate the data density at a rate of $\mathcal{O}(k^{-4/d})$ in KL divergence. For our HRME model, since the general assumptions of these results hold, the uniform convergence also holds.

Table 4.3: Age and Height Estimation Results

Task	Dataset	Method	Male		Female	
			MAE	RMSE	MAE	RMSE
Age	Fisher	SVM-RT	8.83	11.47	8.61	11.17
		NRT	7.20	9.02	6.81	8.53
		HRME-SVR	8.11	11.44	8.09	10.46
		HRME-MLP	**6.91**	**8.74**	**6.40**	**8.07**
Height	SRE	SVM-RT	5.70	7.07	4.85	6.22
		NRT	5.43	6.40	4.27	6.07
		HRME-SVR	5.49	6.89	4.69	6.09
		HRME-MLP	**5.24**	**6.24**	**4.15**	**5.87**

Complexity The complexity of EM-based algorithms for HME models mainly lies in the M-step, where the re-estimation of parameters involves solving a system of equations using the Newton (or Newton-like) update. In the HME models, a Newton iteration cost is $\mathcal{O}(n^3)$. In our case, the complexity of the M-step is in solving the SVM. Specifically, for a standard SVM solver with the primal-dual interior-point method, the complexity is in the Newton update and evaluation of the kernel. Hence, the iteration cost is $\mathcal{O}(n^3 + n^2 d)$. As a result, to attain ϵ-error we need $\mathcal{O}(\epsilon^{-d/2})$ experts. For the HME models, we can assume uniform data partition among experts, and the total cost is $\mathcal{O}(n^3 \epsilon^d)$. For our HRME model, each node's data decreases with depth, and we can take the average among nodes. The resultant total cost is $\mathcal{O}(n^3 \epsilon^d + d n^2 \epsilon^{d/2})$. Although the total complexity increases for our algorithm, the computation can be accelerated using dynamic programming at the price of storage cost. Moreover, the computation at each node can be done in parallel.

Consistency Authors in [34] prove that the least-squares estimators for the ME models are consistent under regularity conditions. Further, authors in [37] show that maximum likelihood estimators are consistent and asymptotically normal. Therefore, our HRME model also produces consistent estimators.

Identifiability Authors in [38] prove that the ME models are identifiable under regularity conditions that the experts are ordered, and the model parameters are

carefully initialized.

In future work, we would like to conduct a more rigorous study of the HRME model's theoretical behaviors.

4.5 Conclusions

This study proposes a hierarchical routing mixture of experts (HRME) model to address the difficulty of partitioning and routing data in conventional regression models. By utilizing a novel gating mechanism that jointly partitions the input-output space, the non-leaf classification experts can separate the modes in the complexly distributed data and route the data to simple leaf regression experts for effective prediction. Furthermore, we develop a probabilistic framework for the HRME model and propose a recursive Expectation-Maximization (EM) based algorithm to optimize the input-output partition, tree structure, and the experts. Comprehensive experimental results on a collection of standard and challenging regression tasks validate our model's effectiveness and highlight some nice properties.

References

[1] W.-Y. Loh. "Fifty years of classification and regression trees". In: *International Statistical Review* 82.3 (2014), pp. 329–348.

[2] L. Breiman. "Random forests". In: *Machine learning* 45.1 (2001), pp. 5–32.

[3] T. L. Bailey, C. Elkan, et al. "Fitting a mixture model by expectation maximization to discover motifs in bipolymers". In: (1994).

[4] R. A. Jacobs et al. "Adaptive mixtures of local experts". In: *Neural computation* 3.1 (1991), pp. 79–87.

[5] S. A. Memon et al. "Neural regression trees". In: *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2019, pp. 1–8.

[6] W. Zhao et al. “Hierarchical routing mixture of experts”. In: *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE. 2021, pp. 7900–7906.

[7] L. Breiman. *Classification and regression trees*. Routledge, 2017.

[8] J. R. Quinlan. “Induction of decision trees”. In: *Machine learning* 1.1 (1986), pp. 81–106.

[9] A. Liaw, M. Wiener, et al. “Classification and regression by randomForest”. In: *R news* 2.3 (2002), pp. 18–22.

[10] S. E. Yuksel, J. N. Wilson, and P. D. Gader. “Twenty years of mixture of experts”. In: *IEEE transactions on neural networks and learning systems* 23.8 (2012), pp. 1177–1193.

[11] M. I. Jordan and R. A. Jacobs. “Hierarchical mixtures of experts and the EM algorithm”. In: *Neural computation* 6.2 (1994), pp. 181–214.

[12] C. M. Bishop and M. Svenskn. “Bayesian hierarchical mixtures of experts”. In: *Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers Inc. 2002, pp. 57–64.

[13] V. Tresp. “Mixtures of Gaussian processes”. In: *Advances in neural information processing systems*. 2001, pp. 654–660.

[14] C. E. Rasmussen and Z. Ghahramani. “Infinite mixtures of Gaussian process experts”. In: *Advances in neural information processing systems*. 2002, pp. 881–888.

[15] C. Yuan and C. Neubauer. “Variational mixture of Gaussian process experts”. In: *Advances in Neural Information Processing Systems*. 2009, pp. 1897–1904.

[16] T. Nguyen and E. Bonilla. “Fast allocation of Gaussian process experts”. In: *International Conference on Machine Learning*. 2014, pp. 145–153.

[17] C. A. Lima, A. L. Coelho, and F. J. Von Zuben. “Hybridizing mixtures of experts with support vector machines: Investigation into nonlinear dynamic systems identification”. In: *Information Sciences* 177.10 (2007), pp. 2049–2074.

[18] L. Cao. “Support vector machines experts for time series forecasting”. In: *Neurocomputing* 51 (2003), pp. 321–339.

[19] S. Waterhouse and A. Robinson. “Pruning and growing hierachical mixtures of experts”. In: (1995).

[20] A. Kanaujia and D. Metaxas. “Learning ambiguities using Bayesian mixture of experts”. In: *Tools with Artificial Intelligence, 2006. ICTAI’06. 18th IEEE International Conference on*. IEEE. 2006, pp. 436–440.

[21] D. Dheeru and E. Karra Taniskidou. *UCI Machine Learning Repository*. 2017. URL: `http://archive.ics.uci.edu/ml`.

[22] P. Tüfekci. “Prediction of full load electrical power output of a base load operated combined cycle power plant using machine learning methods”. In: *International Journal of Electrical Power & Energy Systems* 60 (2014), pp. 126–140.

[23] H. Kaya, P. Tüfekci, and F. S. Gürgen. “Local and global learning methods for predicting power of a combined gas & steam turbine”. In: *Proceedings of the International Conference on Emerging Trends in Computer and Electronics Engineering ICETCEE*. 2012, pp. 13–18.

[24] I.-C. Yeh. “Modeling of strength of high-performance concrete using artificial neural networks”. In: *Cement and Concrete research* 28.12 (1998), pp. 1797–1808.

[25] D. A. Belsley, E. Kuh, and R. E. Welsch. *Regression diagnostics: Identifying influential data and sources of collinearity*. Vol. 571. John Wiley & Sons, 2005.

[26] L. M. Candanedo, V. Feldheim, and D. Deramaix. “Data driven prediction models of energy use of appliances in a low-energy house”. In: *Energy and buildings* 140 (2017), pp. 81–97.

[27] M. Seeger, C. Williams, and N. Lawrence. “Fast forward selection to speed up sparse Gaussian process regression”. In: *Artificial Intelligence and Statistics 9*. EPFL-CONF-161318. 2003.

[28] M. P. Deisenroth and J. W. Ng. “Distributed gaussian processes”. In: *arXiv preprint arXiv:1502.02843* (2015).

[29] L. Buitinck et al. “API design for machine learning software: experiences from the scikit-learn project”. In: *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*. 2013, pp. 108–122.

[30] A. Paszke et al. “Automatic differentiation in PyTorch”. In: (2017).

[31] R. Avnimelech and N. Intrator. “Boosted mixture of experts: an ensemble learning scheme”. In: *Neural computation* 11.2 (1999), pp. 483–497.

[32] D. B. Ferrari and A. Z. Milioni. “Choices and pitfalls concerning mixture-of-experts modeling”. In: *Pesquisa Operacional* 31.1 (2011), pp. 95–111.

[33] M. Trapp et al. “Learning Deep Mixtures of Gaussian Process Experts Using Sum-Product Networks”. In: *arXiv preprint arXiv:1809.04400* (2018).

[34] A. J. Zeevi, R. Meir, and V. Maiorov. “Error bounds for functional approximation and estimation using mixtures of experts”. In: *IEEE Transactions on Information Theory* 44.3 (1998), pp. 1010–1025.

[35] W. Jiang and M. A. Tanner. “On the approximation rate of hierarchical mixtures-of-experts for generalized linear models”. In: *Neural computation* 11.5 (1999), pp. 1183–1198.

[36] W. Jiang and M. A. Tanner. “Hierarchical mixtures-of-experts for generalized linear models: some results on denseness and consistency.” In: *AISTATS*. Citeseer. 1999.

[37] W. Jiang and M. A. Tanner. “On the asymptotic normality of hierarchical mixtures-of-experts for generalized linear models”. In: *IEEE Transactions on Information Theory* 46.3 (2000), pp. 1005–1013.

[38] W. Jiang and M. A. Tanner. “On the identifiability of mixtures-of-experts”. In: *Neural Networks* 12.9 (1999), pp. 1253–1258.

Chapter 5

Data-Specific Models for Speech Feature Discovery

Data-specific models aim to extract intrinsic representations from data that are most informative for VFAH. These representations usually capture patterns that may be hypothesized to be present in the high-dimensional data space, condensed into lower dimensions. The lower-dimensional space is often termed as the *embedding space*, or the *latent space*, and the representations in latent space are called *latent features*. While the latent features are task-agnostic, they are informative and usable in a range of specific tasks.

5.1 Data Assumptions and Latent Feature Discovery

Latent feature discovery is concerned with the automated design or discovery of latent features whose presence is hypothesized but not directly observable or measurable in standard ways. This can be effectively done with data-specific models. To achieve this, we make two basic data assumptions.

Assumption 5.1 (Manifold assumption)**.** Most information in high-dimensional data from the natural world lies on low-dimensional manifolds. Formally, let $\mathcal{M}$ and $\mathcal{N}$ be

M and N dimensional differentiable manifolds, $M < N$. We say that $\mathcal{M}$ is *embedded* in $\mathcal{N}$ if there exists a homeomorphism $f : \mathcal{M} \to \mathcal{N}$ such that for all $p \in \mathcal{M}$, the differential $df_p : T_p\mathcal{M} \to T_{f(p)}\mathcal{N}$ is an injection [1]. Additionally, we may impose metrics on the manifolds $(\mathcal{M}, g)$ and $(\mathcal{N}, h)$ where g and h are Riemannian metrics, i.e., $(0, 2)$-tensors acting as inner products $\langle \cdot, \cdot \rangle_p$ (symmetric, bilinear, positive-definite forms) in the tangent space $T_p\mathcal{M}$ and $T_{f(p)}\mathcal{N}$ for all $p \in \mathcal{M}$. We say the embedding f is *metric-preserving*, or an *isometric embedding*, if for all $p \in \mathcal{M}$ and all $v, w \in T_p\mathcal{M}$, we have $g_p(v, w) = \langle v, w \rangle_p = h_{f(p)} \left(df_p(v), df_p(w) \right) = \langle df_p(v), df_p(w) \rangle_{f(p)}$, i.e., g is the pull-back of h: $g = f^* h$ [1, 2].

Assumption 5.2 (Separability assumption)**.** The distributions of different data sub-classes on a manifold are separable. Let $\mathcal{M}$ be a manifold on which the data lies; $\mathcal{U}$ and $\mathcal{V}$ are subsets of $\mathcal{M}$ with an induced topology where two data sub-classes lie. We impose a probabilistic structure on the subsets $(\mathcal{U}, \Sigma_\mathcal{U}, \mu)$ and $(\mathcal{V}, \Sigma_\mathcal{V}, \nu)$ where $\Sigma_\mathcal{U}$ and $\Sigma_\mathcal{V}$ are Borel σ-algebra compatible with the topology, and μ and ν are probability measures[1] [3, 4]. Define $\mathcal{K}_\mathcal{U} \coloneqq \text{supp}(\mu) \equiv \{E \in \Sigma_\mathcal{U} : \mu(E) \neq 0\}$ and $\mathcal{K}_\mathcal{V} \coloneqq \text{supp}(\nu) \equiv \{F \in \Sigma_\mathcal{V} : \nu(F) \neq 0\}$ such that $\bigcup \mathcal{K}_\mathcal{U} \supseteq \mathcal{U}$ and $\bigcup \mathcal{K}_\mathcal{V} \supseteq \mathcal{V}$, i.e., $\mathcal{K}_\mathcal{U}$ and $\mathcal{K}_\mathcal{V}$ are classes covering $\mathcal{U}$ and $\mathcal{V}$. Then we have $\mathcal{K}_\mathcal{U} \bigcup \mathcal{K}_\mathcal{V} = \mathcal{U} \bigcup \mathcal{V}$ and $\mathcal{K}_\mathcal{U} \bigcap \mathcal{K}_\mathcal{V} = \emptyset$.

The two assumptions above are schematically illustrated in Figure 5-1. Based on these assumptions, we can build data-specific models that can (1) encode high-dimensional data into lower dimensions and (2) ensure that the low-dimensional encodings (representations) from different sub-classes are separable. Consequently, we obtain a latent space that admits a separable structure of the latent features from different sub-classes. The latent features in each sub-class are associated with a subset of profiling parameters from the data. From this point of view, these latent features are *disentangled*. They can be used in specific VFAH tasks, such as classifying or predicting profiling parameters.

[1]The Borel σ-algebra is used to assign probabilities to the open sets in the topology.

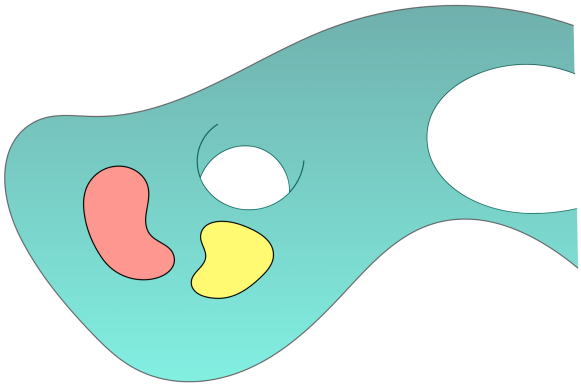


Figure 5-1: Illustration of manifold data assumptions. The data lies on a lower-dimensional manifold, where the two data sub-classes (red and yellow) are separable by topology.

5.2 Discovering Separable Latent Features with Generative Models

To extract latent features from data, one needs models that satisfy the two requirements introduced above. Many generative models satisfy the first requirement only—the manifold requirement. They can "condense" high dimensional data distributions into lower-dimensional representations and then generate samples of the same distribution from these representations. For example, graphical generative models such as principle component analysis (PCA) [5], Gaussian mixture models (GMM) [5], hidden Markov model (HMM) [6, 7], Bayesian nets, Markov random fields [8, 9], restricted Boltzmann machines (RBM) [10] etc, represent data distributions via low-dimensional sufficient statistics or hidden states. On the other hand, deep generative models such as deep belief nets (DBN) [11, 12, 13], variational autoencoders (VAE) [14], generative adversarial nets (GAN) [15, 16, 17] contain data distribution information in their hidden neurons or latent encoding. Among these models, GANs attract intensive interest from the research community due to their ability to generate samples (especially images) with intricate detail and remarkable fidelity.

For these models, while the first requirement of a low-dimensional manifold enabling latent feature discovery is easy to achieve, the second requirement of separability is not. The separability of the latent representations presents a non-trivial problem in this

context. The separability of latent features is related to the "disentanglement" in the representation learning literature. From the perspective of VFAH, one can view this as the problem of disambiguating the influences of different profiling parameters on voice by designing non-confounding features. Several deep models have been proposed for latent feature disentanglement. Among the deep generative models, many attempts to learn low-dimensional multivariate latent variables that semantically correlate with the observations in an unsupervised manner [18, 19, 20, 21, 22] have been made. However, well-disentangled representations are difficult to find in an unsupervised manner [23]. To understand this, let us first formally define disentanglement (a weaker requirement than separability).

Definition 5.1 (Disentangled latent representation)**.** On the latent manifold, denote the underlying subset of a data sub-class $\mathcal{U}$ as $\underline{\mathcal{U}}$. For each data $x \in \mathcal{U}$ and a latent representation $z \in \underline{\mathcal{U}}$, there exists a local diffeomorphism $G : \underline{U} \to U$ where $\underline{U}$ and U are neighborhoods of z and x, and hence the differential $d_z G : T_z\underline{\mathcal{U}} \to T_x\mathcal{U}$ is a linear isomorphism. In finite-dimensional space, this means the Jacobian $J_z G$ is non-singular. Then z is a disentangled latent representation for the data x.

In this way, a variation in the latent dimension independently leads to a corresponding variation in the original data space [24]. However, the problem is that the latent representations may not be identifiable, i.e., they may have the same distribution. Stating formally

Definition 5.2 (Unidentifiable latent representation)**.** Consider a latent manifold equipped with a probability measure $(\underline{\mathcal{U}}, \Sigma_{\underline{\mathcal{U}}}, \underline{\mu})$. For every latent representations $w \in \underline{\mathcal{U}}$, there exists a measure-preserving map $g : \underline{\mathcal{U}} \to \underline{\mathcal{U}}$ (such as change of coordinate) such that $\underline{\mu}(\{u \mid u \leq g(w)\}) = g^*\underline{\mu}(\{u \mid u \leq w\})$ where $g^*\underline{\mu}$ is the pull-back of $\underline{\mu}$ [25]. Therefore, two latent representations can have the same probability and are unidentifiable up to isomorphism.

This motivates us to adopt a supervised or semi-supervised approach and use a supervision signal to identify latent representations on the latent manifold while keeping them disentangled. To this end, many supervised or semi-supervised deep

generative models have been proposed, such as [26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37].

Among these models, one school of methods attempts to impose geometric structures onto the latent space, such as multi-modal distributions, orthogonal subspaces, and Riemannian manifold [34, 35, 36, 38, 39, 40]. These methods are of particular interest because they disentangle and further enforce the separability constraint on the latent manifold, and separability is a stronger property than disentanglement.

Our study pursues this school of geometric approaches. We propose a semi-supervised generative model to enforce disentanglement and geometric separability of latent representations. Further, we impose an algebraic structure on the latent manifold to allow vector-space arithmetic operations, which have semantic interpretations in the data (observation) space. In the context of VFAH, we would like to discover latent features that represent the information in the voice and are also separable w.r.t. classes such as gender, dialect, emotions, etc. In other words, they are disambiguated. To this end, we propose a class-dependent adversarial latent structure matching (CALM) framework. We describe this framework in detail in the following section.

5.3 Automatic Speech Feature Discovery via Class-Dependent Adversarial Latent Structure Matching

Having discussed the general approaches to learning latent features using generative models, we present an adversarial modeling and learning approach to discovering speech features.

5.3.1 Introduction

As speech technology advances, deriving features for characterizing domain-specific explanatory traits in speech remains an active and challenging task [41]. The domain-specific speech features are mainly derived from spectrograms, including cepstral

representations such as the Mel-frequency cepstral coefficients [42], statistical representations such as the Gaussian mixture models [43], subspace representations such as the i-vectors [44], and lately developed latent representations such as neural networks [45]. These domain-specific features have proven their effectiveness in various speech-related tasks [46].

On the other hand, the speech spectrogram is a powerful representation of the time and frequency information in voice production, transmission, and acquisition, capable of characterizing both biometric and environmental traits [47]. More specifically, the variations in spectrogram are underpinned by a set of profiling parameters, including physical parameters such as unique articulatory apparatus configurations and body shape, physiological parameters such as age and illness, psychological parameters such as mental states and emotions, sociological parameters such as race and education level, environmental parameters such as location and surroundings, etc. [48]. These parameters embody high-resolution, fine-detail signatures in time and frequency in the spectrogram and can be used to predict speaker traits. For instance, the harmonic bandwidth can predict age [49], the voicing onset time can identify Parkinson's disease [50], the variations in formant characteristics can help break voice disguise [48], etc.

Generally, such signature variations are detected through human observation as patterns within quantifiable entities such as formants, jitter, shimmer, etc. However, not all profiling parameters may necessarily be present or well-disambiguated in spectrograms. They may be highly transient and micro in terms of distinguishability and may measurably emerge in higher-dimensional space. Further, we may not be able to disentangle the correspondence of different parameters with signature variations as-is.

This study addresses the problem of automatic speech feature discovery by connecting the time-spectral domain and the latent feature domain. Specifically, we build a bi-directional mapping between the spectrogram and its latent space representation and impose a fully parametric class-dependent geometry onto the latent space via adversarial matching. This provides a semantic link between latent representations

Table 5.1: Symbol List

Symbol	Description
$\mathcal{X}$	original spectral space
$\mathcal{Y}$	original label space
$\widehat{\mathcal{X}}$	reconstructed/generated spectral space
$\mathcal{Z}$	latent space
$\widetilde{\mathcal{Z}}$	latent space with prior structure
$\mathbb{P}_s$	Gaussian distributions on $\widetilde{\mathcal{Z}}$: $\mathbb{P}_s = \{\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \mid \boldsymbol{\mu}_i \in \mathbb{R}^L, \boldsymbol{\Sigma}_i \in \boldsymbol{\Lambda}^L\}_{i=1}^C$
f_θ	encoding map $f_\theta : \mathcal{X} \to \mathcal{Z}$
g_ψ	decoding/generating map $g_\psi : \mathcal{Z} \to \widehat{\mathcal{X}}$
h_ζ	preconditioning map $h_\zeta : \mathcal{Y} \to \widetilde{\mathcal{Z}}$
m_ϕ	discriminating map $m_\phi : \mathcal{Z} \cup \widetilde{\mathcal{Z}} \to \mathbb{R}$
u	probability measure on $\mathcal{X}$
v	probability measure on $\mathcal{Z}$
$\widetilde{v}$	probability measure on $\widetilde{\mathcal{Z}}$
$\mathcal{W}(\cdot, \cdot)$	Wasserstein distance

and spectrograms, enabling semantic operations such as sampling, interpolation, and reconstruction to be performed on them. Moreover, the latent space admits a natural clustering, enabling direct classification. We call our approach class-dependent adversarial latent structure matching (CALM).

5.3.2 Class-Dependent Adversarial Latent Structure Matching

This section proposes the class-dependent adversarial latent structure matching (CALM) framework. Consider a collection of data $\mathcal{D} = \mathcal{X} \times \mathcal{Y} = \{(\mathbf{X}_i, y_i)\}_{i=1}^N$ with N samples from C classes, where $\mathbf{X}_i$ represents the spectrogram in class $y_i \in [1, \ldots, C]$. Assume the underlying distribution for $\mathcal{X}$ is $\mathbb{P}_\mathcal{X}$ with probability measure u, and the underlying distribution for the latent space $\mathcal{Z} \in \mathbb{R}^L$ of dimension L is $\mathbb{P}_\mathcal{Z}$ with probability measure v. First, define the encoding map between $\mathcal{X}$ and $\mathcal{Z}$ as $f_\theta : \mathcal{X} \to \mathcal{Z}$, and the decoding/reconstructing/generating map as $g_\psi : \mathcal{Z} \to \widehat{\mathcal{X}}$ where $\widehat{\mathcal{X}}$ is the reconstructed/generated data space. The encoder (E) f_θ and generator (G) g_ψ are in the form of deterministic universal approximators (e.g., a multi-layer neural networks)

parameterized by θ and ψ respectively. In order to encode sufficient information of $\mathcal{X}$ in $\mathcal{Z}$ such that one can reconstruct $\mathbf{X} \in \mathcal{X}$ from $\mathbf{z} \in \mathcal{Z}$, we minimize the difference between $\mathcal{X}$ and $\widehat{\mathcal{X}}$:

$$\min_{\psi} \mathcal{L}_{\text{reconstruction}} \coloneqq \mathbb{E}_{\mathbf{X}\sim u, \mathbf{z}\sim v} \|\mathbf{X} - g_{\psi}(\mathbf{z})\|_F^2 \tag{5.1}$$

where $\|\cdot\|_F$ is the Frobenius norm.

Second, we impose the latent space $\mathcal{Z}$ with a desirable structure by matching it to a prior $\widetilde{\mathcal{Z}}$, which carries a class-dependent distribution $\mathbb{P}_s$ with probability measure $\widetilde{v}$. The choice of the prior distribution controls the information encoded in the latent space. Studies show that imposing a Gaussian prior to the latent space realizes global and local information decomposition [27]. We further show that by imposing a class-dependent Gaussian prior, the latent vectors in the latent space naturally cluster into different classes and, therefore, can be directly used for classification tasks. For instance, if the prior depends on speaker classes, we can deduce a voice sample's speaker id by finding the prior class having the smallest distance (defined by the geodesic on the latent manifold) to its latent vector. Hence, we formulate $\mathbb{P}_s$ as a collection of Gaussians $\mathbb{P}_s = \{\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \mid \boldsymbol{\mu}_i \in \mathbb{R}^L, \boldsymbol{\Sigma}_i \in \boldsymbol{\Lambda}^L\}_{i=1}^C$ where $\boldsymbol{\Lambda}^L$ denotes the group of L-by-L diagonal matrices. We train a preconditioner (P) network $h_{\zeta} : \mathcal{Y} \to \widetilde{\mathcal{Z}}$ to transform class labels into prior vectors. Additionally, we adopt the reparameterization trick [14] to avoid computing stochastic gradients in the network

$$\widetilde{\mathbf{z}}_i \sim \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \Rightarrow \widetilde{\mathbf{z}}_i = \boldsymbol{\mu}_i + \epsilon\sqrt{\boldsymbol{\Sigma}_i},\ \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{1}) \tag{5.2}$$

After reparameterization, the preconditioner produces the mean $\boldsymbol{\mu}$ and variance $\boldsymbol{\Sigma}$ for each $y \in \mathcal{Y}$, and randomly samples from standard normal distribution to produce $\widetilde{\mathbf{z}}$. Next, we impose a separable structure to the Gaussians $\mathbb{P}_s$ by minimizing the scatter

loss

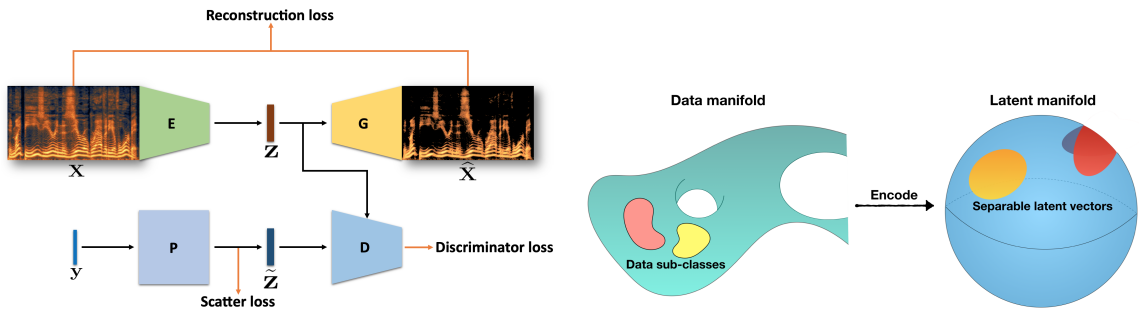
$$\begin{aligned}\min \mathcal{L}_{\text{scatter}} \coloneqq & \left| \frac{1}{C} \sum_{i=1}^{C} \|\boldsymbol{\mu}_i\| - \sqrt{C} \right| + \\ & \sum_{i=1}^{C} \left| \|\boldsymbol{\Sigma}_i\| + \frac{\sqrt{C}}{2} - \|\boldsymbol{\mu}_i\| \right| + \\ & \sum_{i=1}^{C-1} \sum_{j=i+1}^{C} \left| \left\langle \frac{\boldsymbol{\mu}_i}{\|\boldsymbol{\mu}_i\|}, \frac{\boldsymbol{\mu}_j}{\|\boldsymbol{\mu}_j\|} \right\rangle \right| \end{aligned} \tag{5.3}$$

where the first term constrains the mean vectors onto a hyper-sphere with diameter $\sqrt{C}$, the second term is a constraint on within-class scatter relative to between-class scatter, and the third term is the relative scatter of mean vectors. Having defined the separable, class-dependent distribution structure on the prior $\widetilde{\mathcal{Z}}$, now we can pull the latent space $\mathcal{Z}$ close to $\widetilde{\mathcal{Z}}$ via adversarial matching. This is achieved by minimizing the approximation error between v and $\widetilde{v}$, i.e., minimizing their Wasserstein distance [16]

$$\min \ \mathcal{L}_{\text{Wasserstein}} \coloneqq \mathcal{W}(v, \widetilde{v}) = \inf_{\gamma \in \prod(v, \widetilde{v})} \mathbb{E}_{(\mathbf{z}, \widetilde{\mathbf{z}}) \sim \gamma} \|\mathbf{z} - \widetilde{\mathbf{z}}\|_1 \tag{5.4}$$

where $\prod(v, \widetilde{v})$ denotes the set of joint probability measures $\gamma(\mathbf{z}, \widetilde{\mathbf{z}})$ whose marginals are respectively v and $\widetilde{v}$, and $\|\cdot\|$ denotes the l_1 norm. In [16] it shows that $\mathcal{W}(v, \widetilde{v}) \to 0$ implies $v \xrightarrow{d} \widetilde{v}$. To minimize (5.4), we train a discriminator (D) $m_\phi : \mathcal{Z} \cup \widetilde{\mathcal{Z}} \to \mathbb{R}$ in the family of 1-Lipschitz continuous functions. The discriminator reaches its optimum when v matches with $\widetilde{v}$ [16]. The discriminator makes equal errors identifying samples from $\mathcal{Z}$ or $\widetilde{\mathcal{Z}}$.

In addition to the separability of latent features, our CALM framework adds an algebraic structure to the latent space. From the tangent space of the latent manifold, vector space operations such as vector addition (translation) and scalar multiplication (scaling) can be performed on the latent vectors. Consequently, interpolation can be done in the latent space. Since we have regularized the latent manifold to be spherical, its geodesics are the great circles through the center. By spherically interpolating along the geodesic on the hyper-sphere, we can generate a smooth transition of samples in the sample space. Hence, the algebraic operations in the latent space have corresponding



(a) (b)

Figure 5-2: CALM framework. **(a)** The class-dependent adversarial latent matching model. It consists of an encoder E, a decoder G, a preconditioner P, and a discriminator D. **(b)** The E, G, P, and D together project the data sub-classes onto separable sub-manifolds in the latent manifold. The E, G pair encodes necessary information in the latent manifold for reconstruction, P enforces class-dependent separable distribution constraint, and the game between D and G ensures the sub-classes are mapped to corresponding sub-manifolds.

semantic interpretations in the sample space.

The proposed CALM framework with components E, G, P, and D is shown in Figure 5-2a. The encoder-decoder pair (E, G) encodes the speech's spectrograms (or other data representations) into latent features in the latent space. The preconditioner P transforms class labels (referring to classes such as gender, dialect, and emotion) into a prior distribution of non-overlapping Gaussians over the latent manifold. This can be done by transforming the labels through a neural net into mean and variance vectors using the reparameterization trick. The collection of Gaussians is constrained to be evenly distributed on a hyper-sphere (as illustrated in Figure 5-2b), with their centers scattered and their spread confined by the scatter loss (5.3). The number of Gaussians is equal to the number of classes. The discriminator tries to match the latent feature distribution with the prior; the latent features are separable and class-dependent.

To jointly optimize over objective (5.1), (5.3), and (5.4), we propose a three-phase iterative training comprising an encoding phase, discriminating phase, and adversarial phase (see Figure 5-3). In phase one, the encoder produces encoding $\mathbf{z}$ by minimizing (5.1), and the preconditioner elects prior encoding $\tilde{\mathbf{z}}$ by minimizing (5.3). In phase two, the discriminator classifies between $\mathbf{z}$ and $\tilde{\mathbf{z}}$ by minimizing (5.4). In phase three, the encoder and preconditioner adjust themselves by minimizing the reverse of (5.4),

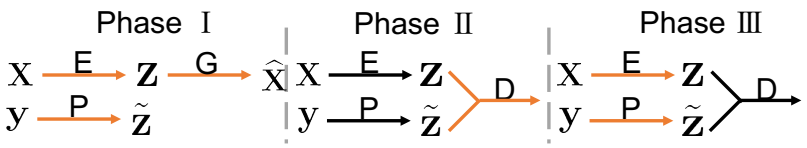


Figure 5-3: Three-phase training of CALM.

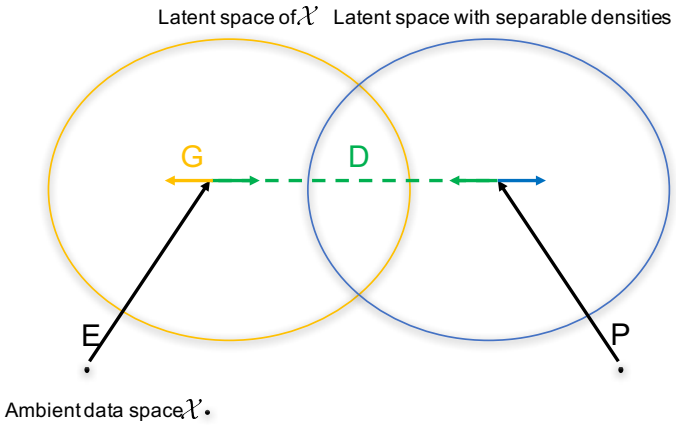


Figure 5-4: The game among encoder E, decoder G, preconditioner P and discriminator D in CALM.

which is an adversarial process [15].

The three-phase learning can be viewed as a game among E, G, P, D (as illustrated in Figure 5-4): the encoder and preconditioner projects data to the latent space, and the latent space with separable densities, respectively; then the discriminator measures the difference between the two latent spaces and pulls them toward each other following the gradient flow; meanwhile, the generator fights against the discriminator by pulling the latent spaces in opposite directions.

5.3.3 Experiments

This section describes the experiments we conduct to validate the effectiveness of the CALM framework. We demonstrate some nice properties of the latent space learned by our CALM framework and the utility of the learned features in VFAH classification and regression tasks.

Experimental Settings

Task and Data We evaluate our CALM framework by (1) reconstructing spectrograms from latent features, (2) sampling latent space across classes, (3) sampling latent space within a class, and (4) performing various classification and regression tasks using the latent features, including the deduction of gender, dialect, age, and height from speech.

For gender and dialect classification, we use the TIMIT dataset [51], which consists of 6300 recordings with ten sentences spoken by 630 speakers in eight major dialect regions of the United States. The recordings are first chunked into 2-second segments and then converted to constant-Q spectrograms [52] of dimension 414×450 to leverage pitch variations. Eighty percent of the data is used for training, and the rest is used for testing. The data are normalized to zero mean and identity covariance. For age and height estimation, we use the task and data settings described in Section 3.2.3, except that instead of extracting i-vector features, we compute the constant-Q features.

Network Structure The encoder E has five blocks of {convolutional layer, batch normalization, leaky ReLU activation} with filter size 4, stride size 2, padding size 1 and number of filters {64, 128, 256, 512, 1024}, respectively. It outputs a 200 dimensional latent vector via average pooling. The generator G has five blocks of {transposed convolutional layer, batch normalization, leaky ReLU activation} with filter size 4, stride size 2, padding size 1 and number of filters {1024, 512, 256, 128, 64}, respectively. It outputs the spectrogram via max pooling. The preconditioner P has three fully connected layers with size 1000, batch normalization, and leaky ReLU activation. The discriminator D has three fully connected layers with size 1000, leaky ReLU activation, and additional gradient penalty to ensure the 1-Lipschitz continuity [53].

Training We use the proposed three-phase training. We use an ADAM optimizer with an initial learning rate of 0.0001 and batch size of 4.

Latent Reconstruction and Sampling

Figure 5-5 shows the reconstructed spectrograms for different genders and dialects on the test data. It demonstrates high-quality reconstruction with continuous harmonics and fine details, indicating that the latent features capture class-dependent information. Figure 5-6 shows the samples generated from the spherical sampling of the latent space from speaker A to speaker B [54]. It demonstrates a smooth transition between the two speakers and empirically suggests that the latent manifold admits a spherical structure and supports algebraic-semantic operations. Figure 5-7 shows the spectrograms generated from sampling the latent submanifolds for different genders and dialects. It demonstrates strong class dependency and within-class variations, implying that the latent manifold has class-dependent submanifolds.

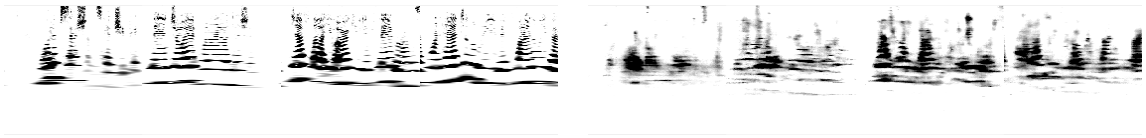
Classification and Regression with Latent Features

Further, we test the learned latent space with gender and dialect classification. For a test spectrogram $\mathbf{X}_t$, we encode it to its latent representation $\mathbf{z}_t$ and select the class label with the smallest Mahalanobis distance to the prior distribution: label $= \arg\min_i dist(\mathbf{z}_t, \{\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)\}_{i=1}^{C})$. We obtain 76% accuracy for gender classification and 72% for dialect classification. This empirically verifies that the latent space observes a class-dependent clustering structure. It is worth noting that we do not train a classifier but rather implicitly incorporate the class information into the latent space. Classification accuracy can be improved by imposing a more refined separated structure or employing an additional classifier (e.g., neural nets) to classify the latent features. To this end, we train a two-layer feedforward network on the latent features obtained from our CALM framework and obtain 99% gender classification accuracy and 97% dialect classification accuracy.

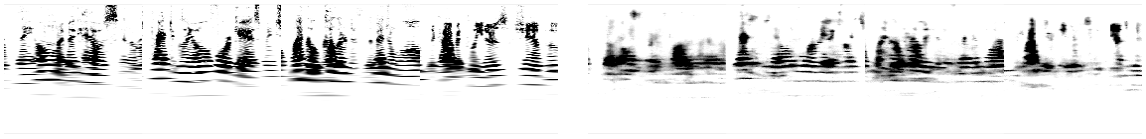
For age and height estimation, we need to adapt our CALM framework to a regression setting. Specifically, we first train our CALM models conditioned on speaker id—using speaker ids as inputs to the preconditioner. Consequently, the learned latent space carries a speaker-class-dependent structure (though it is not refined enough for

Table 5.2: Age and Height Estimation Results

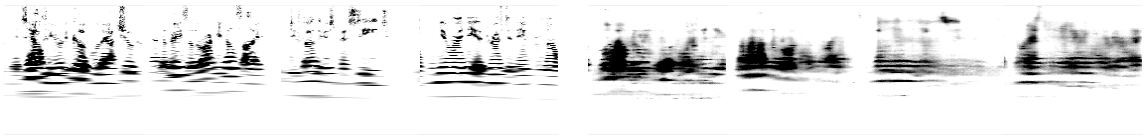
Task	Dataset	Method	Male		Female	
			MAE	RMSE	MAE	RMSE
Age	Fisher	NRT	7.20	9.02	6.81	8.53
		HRME-MLP	**6.91**	**8.74**	**6.40**	**8.07**
		CALM-MLP	7.28	9.64	7.25	9.58
Height	SRE	NRT	5.43	6.40	4.27	6.07
		HRME-MLP	**5.24**	**6.24**	**4.15**	**5.87**
		CALM-MLP	5.95	6.53	4.88	6.18



(a)



(b)



(c)

Figure 5-5: Within-class spectrogram reconstruction: **(a)** original and reconstructed females, **(b)** original and reconstructed males, **(c)** original and reconstructed dialects.

direct use in identifying speakers). Then, we train a two-layer feedforward network on the learned latent features to predict age and height.

Table 5.2 shows the results for age and height estimation. The NRT and HRME results are taken from Section 3.2.3 and 4.3.3, respectively. We observe that MLP with CALM features achieves comparable performance to NRT with i-vector features. Considering that the CALM features are conditioned on speaker ids, we validate that the CALM feature space embeds a speaker-dependent structure potentially similar to the i-vector subspace structure.

Figure 5-6: Generated spectrograms via sampling the latent space from speaker A to speaker B.

5.3.4 Related Work

Our work adopts the idea of adversarial matching of two distributions and the adversarial autoencoder structure [27, 37]. The adversarial learning is originally proposed in [15], and then variants arise to improve the generalization ability and stability [16, 55]. The generative adversarial net is mainly used to generate samples close to the original data domain [27] or transfer them to a different domain [56, 57, 58].

One challenging task in adversarial learning is interpreting the relation between the data domain and the latent domain [59, 60, 61]. Some recent work uses the latent domain to improve generalization in data domain [62] or cross domains [57, 58], or to decompose information in the data domain [27]. While related, our work takes a very different path by imposing a class-dependent structure on the latent domain, whereas similar work uses standard normal [16] or categorical manifolds [27, 37]. One close work [63] sets the latent space to be a unit ball by re-scaling random vectors drawn from the Gaussian but does not account for class dependency. By enforcing the latent structure prior to being class-dependent, we obtain a semantic interpretation and manipulation of the latent space.

5.3.5 Conclusions

This study presents a class-dependent adversarial latent structure matching (CALM) framework to encode a class-dependent separable structure into the latent space. The latent features in the latent space fall into natural clusters and can be directly used for classification tasks. The latent space also admits an algebraic structure that allows sampling and interpolation within/across classes with semantic interpretation. Hence, our CALM framework provides a semantic link between the latent and data space

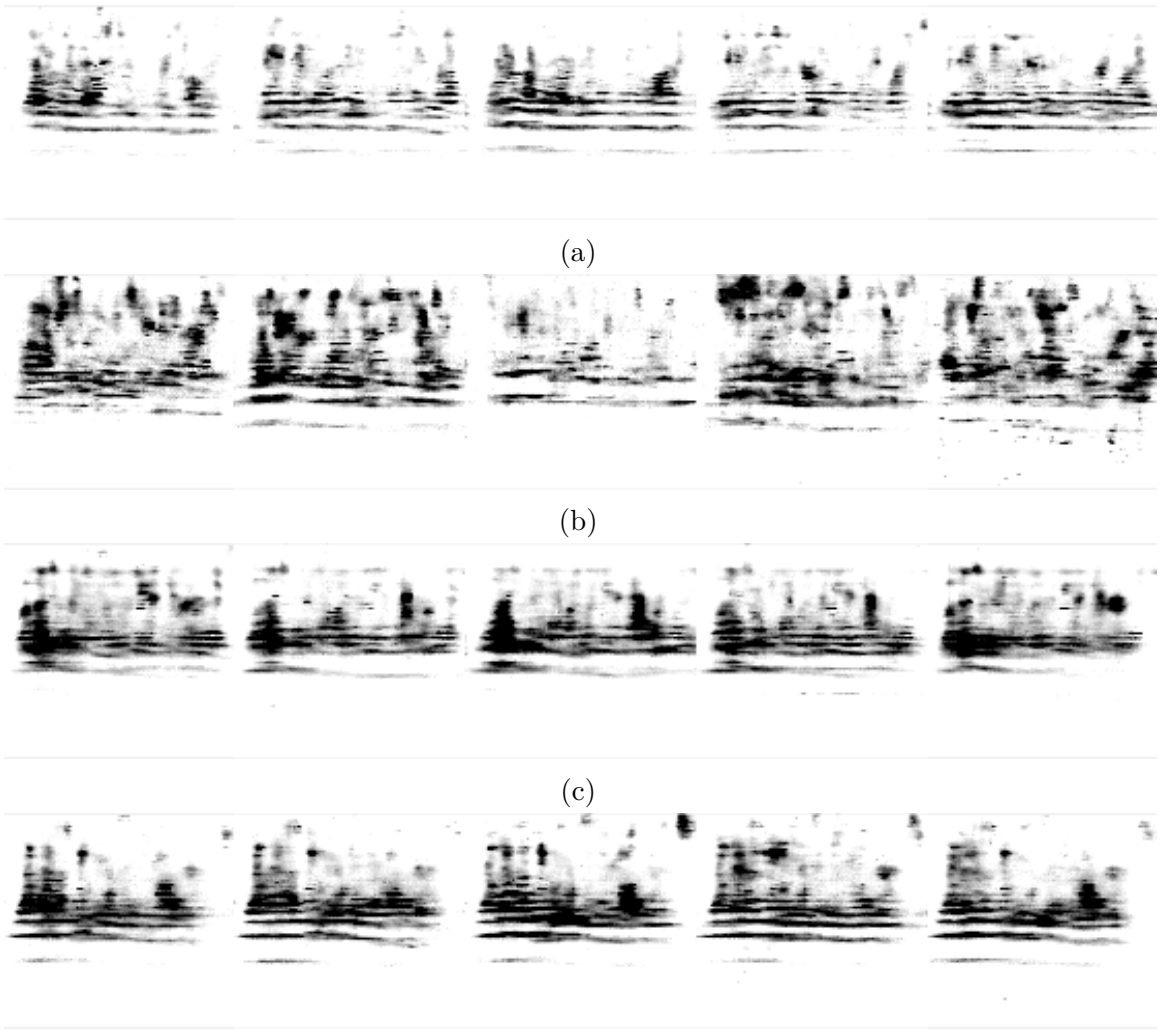


Figure 5-7: Generated spectrograms via sampling the latent space for **(a)** female, **(b)** male, **(c)** and **(d)** dialect 1 and 2.

and a tool to discover and analyze latent features. The effectiveness of our framework is validated through various VFAH classification and regression tasks. Future work could further study the structure in latent space, such as via subspace decomposition, and improve the stability of the adversarial matching. Another research direction is to explore the utility of generated samples to augment data and improve task performance.

References

[1] M. P. Do Carmo and J. Flaherty Francis. *Riemannian geometry*. Vol. 6. Springer, 1992.

[2] S. Lang. *Introduction to differentiable manifolds*. Springer Science & Business Media, 2006.

[3] V. I. Bogachev and M. A. S. Ruas. *Measure theory*. Vol. 1. Springer, 2007.

[4] P. R. Halmos. *Measure theory*. Vol. 18. Springer, 2013.

[5] C. M. Bishop. *Pattern recognition and machine learning*. springer, 2006.

[6] L. R. Rabiner. "A tutorial on hidden Markov models and selected applications in speech recognition". In: *Proceedings of the IEEE* 77.2 (1989), pp. 257–286.

[7] J. A. Bilmes et al. "A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models". In: *International Computer Science Institute* 4.510 (1998), p. 126.

[8] D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[9] M. I. Jordan. *Learning in graphical models*. Vol. 89. Springer Science & Business Media, 1998.

[10] R. Salakhutdinov and G. Hinton. "Deep boltzmann machines". In: *Artificial intelligence and statistics*. 2009, pp. 448–455.

[11] G. E. Hinton, S. Osindero, and Y.-W. Teh. "A fast learning algorithm for deep belief nets". In: *Neural computation* 18.7 (2006), pp. 1527–1554.

[12] K. Swersky et al. "A tutorial on stochastic approximation algorithms for training restricted Boltzmann machines and deep belief nets". In: *2010 Information Theory and Applications Workshop (ITA)*. IEEE. 2010, pp. 1–10.

[13] A.-r. Mohamed, G. E. Dahl, and G. Hinton. "Acoustic modeling using deep belief networks". In: *IEEE transactions on audio, speech, and language processing* 20.1 (2011), pp. 14–22.

[14] D. P. Kingma and M. Welling. “Auto-encoding variational bayes”. In: *arXiv preprint arXiv:1312.6114* (2013).

[15] I. Goodfellow et al. “Generative adversarial nets”. In: *Advances in neural information processing systems*. 2014, pp. 2672–2680.

[16] M. Arjovsky, S. Chintala, and L. Bottou. “Wasserstein generative adversarial networks”. In: *International conference on machine learning*. 2017, pp. 214–223.

[17] A. Radford, L. Metz, and S. Chintala. “Unsupervised representation learning with deep convolutional generative adversarial networks”. In: *arXiv preprint arXiv:1511.06434* (2015).

[18] X. Li et al. “SCGAN: Disentangled Representation Learning by Adding Similarity Constraint on Generative Adversarial Nets”. In: *IEEE Access* (2018).

[19] K. K. Singh, U. Ojha, and Y. J. Lee. “FineGAN: Unsupervised Hierarchical Disentanglement for Fine-Grained Object Generation and Discovery”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 6490–6499.

[20] T. Karras, S. Laine, and T. Aila. “A style-based generator architecture for generative adversarial networks”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 4401–4410.

[21] J. Oldfield, Y. Panagakis, and M. A. Nicolaou. “Adversarial Learning of Disentangled and Generalizable Representations for Visual Attributes”. In: *arXiv preprint arXiv:1904.04772* (2019).

[22] W. Wu et al. “Disentangling Content and Style via Unsupervised Geometry Distillation”. In: *arXiv preprint arXiv:1905.04538* (2019).

[23] F. Locatello et al. “Challenging common assumptions in the unsupervised learning of disentangled representations”. In: *arXiv preprint arXiv:1811.12359* (2018).

[24] Y. Bengio, A. Courville, and P. Vincent. "Representation learning: A review and new perspectives". In: *IEEE transactions on pattern analysis and machine intelligence* 35.8 (2013), pp. 1798–1828.

[25] R. J. McCann et al. "Existence and uniqueness of monotone measure-preserving maps". In: *Duke Mathematical Journal* 80.2 (1995), pp. 309–324.

[26] M. Tschannen, O. Bachem, and M. Lucic. "Recent advances in autoencoder-based representation learning". In: *arXiv preprint arXiv:1812.05069* (2018).

[27] A. Makhzani and B. J. Frey. "Pixelgan autoencoders". In: *Advances in Neural Information Processing Systems*. 2017, pp. 1975–1985.

[28] I. Tolstikhin et al. "Wasserstein auto-encoders". In: *arXiv preprint arXiv:1711.01558* (2017).

[29] J. J. Zhao et al. "Adversarially Regularized Autoencoders." In: *ICML*. 2018, pp. 5897–5906.

[30] L. Ma et al. "Disentangled person image generation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 99–108.

[31] L. Tran, X. Yin, and X. Liu. "Disentangled representation learning gan for pose-invariant face recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 1415–1424.

[32] A. Gonzalez-Garcia, J. van de Weijer, and Y. Bengio. "Image-to-image translation for cross-domain disentanglement". In: *Advances in Neural Information Processing Systems*. 2018, pp. 1287–1298.

[33] Y. Liu et al. "Multi-task adversarial network for disentangled feature learning". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 3743–3751.

[34] L. Tran et al. "Disentangling Geometry and Appearance with Regularised Geometry-Aware Generative Adversarial Networks". In: *International Journal of Computer Vision* 127.6-7 (2019), pp. 824–844.

[35] Z. Zheng and L. Sun. "Disentangling Latent Space for VAE by Label Relevant/Irrelevant Dimensions". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 12192–12201.

[36] A. Shukla et al. "Product of Orthogonal Spheres Parameterization for Disentangled Representation Learning". In: *arXiv preprint arXiv:1907.09554* (2019).

[37] A. Makhzani et al. "Adversarial autoencoders". In: *arXiv preprint arXiv:1511.05644* (2015).

[38] A. Shukla et al. "Geometry of Deep Generative Models for Disentangled Representations". In: *arXiv preprint arXiv:1902.06964* (2019).

[39] M. Awiszus, H. Ackermann, and B. Rosenhahn. "Learning disentangled representations via independent subspaces". In: *arXiv preprint arXiv:1908.08989* (2019).

[40] I. Ovinnikov. "Poincar\'e Wasserstein Autoencoder". In: *arXiv preprint arXiv:1901.01427* (2019).

[41] U. Shrawankar and V. M. Thakare. "Techniques for feature extraction in speech recognition system: A comparative study". In: *arXiv preprint arXiv:1305.1145* (2013).

[42] S. Davis and P. Mermelstein. "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences". In: *IEEE transactions on acoustics, speech, and signal processing* 28.4 (1980), pp. 357–366.

[43] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn. "Speaker verification using adapted Gaussian mixture models". In: *Digital signal processing* 10.1 (2000), pp. 19–41.

[44] N. Dehak et al. "Front-end factor analysis for speaker verification". In: *IEEE Transactions on Audio, Speech and Language Processing* 19.4 (2011), pp. 788–798.

[45] Z. Wu et al. “Deep neural networks employing multi-task learning and stacked bottleneck features for speech synthesis”. In: *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE. 2015, pp. 4460–4464.

[46] J. H. Hansen and T. Hasan. “Speaker recognition by machines and humans: A tutorial review”. In: *IEEE Signal processing magazine* 32.6 (2015), pp. 74–99.

[47] R. Singh, J. Keshet, and E. Hovy. “Profiling hoax callers”. In: *Technologies for Homeland Security (HST), 2016 IEEE Symposium on*. IEEE. 2016, pp. 1–6.

[48] R. Singh, B. Raj, and D. Gencaga. “Forensic anthropometry from voice: an articulatory-phonetic approach”. In: *2016 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. IEEE. 2016, pp. 1375–1380.

[49] J. F. Lehman and R. Singh. “Estimation of Children’s Physical Characteristics from Their Voices.” In: *INTERSPEECH*. 2016, pp. 1417–1421.

[50] E. Fischer and A. M. Goberman. “Voice onset time in Parkinson disease”. In: *Journal of Communication Disorders* 43.1 (2010), pp. 21–34.

[51] V. Zue, S. Seneff, and J. Glass. “Speech database development at MIT: TIMIT and beyond”. In: *Speech communication* 9.4 (1990), pp. 351–356.

[52] J. C. Brown. “Calculation of a constant Q spectral transform”. In: *The Journal of the Acoustical Society of America* 89.1 (1991), pp. 425–434.

[53] I. Gulrajani et al. “Improved training of wasserstein gans”. In: *arXiv preprint arXiv:1704.00028* (2017).

[54] T. White. “Sampling generative networks: Notes on a few effective techniques”. In: *arXiv preprint arXiv:1609.04468* (2016).

[55] B. Neyshabur, S. Bhojanapalli, and A. Chakrabarti. “Stabilizing GAN Training with Multiple Random Projections”. In: *arXiv preprint arXiv:1705.07831* (2017).

[56] Z. Yi, H. Zhang, P. T. Gong, et al. “DualGAN: Unsupervised Dual Learning for Image-to-Image Translation”. In: *arXiv preprint arXiv:1704.02510* (2017).

[57] J.-Y. Zhu et al. “Unpaired image-to-image translation using cycle-consistent adversarial networks”. In: *arXiv preprint arXiv:1703.10593* (2017).

[58] T. Kim et al. “Learning to discover cross-domain relations with generative adversarial networks”. In: *arXiv preprint arXiv:1703.05192* (2017).

[59] Z. C. Lipton and S. Tripathi. “Precise Recovery of Latent Vectors from Generative Adversarial Networks”. In: *arXiv preprint arXiv:1702.04782* (2017).

[60] J. Luo. “Learning Inverse Mapping by Autoencoder based Generative Adversarial Nets”. In: *arXiv preprint arXiv:1703.10094* (2017).

[61] X. Chen et al. “Infogan: Interpretable representation learning by information maximizing generative adversarial nets”. In: *Advances in Neural Information Processing Systems*. 2016, pp. 2172–2180.

[62] J. Donahue, P. Krähenbühl, and T. Darrell. “Adversarial feature learning”. In: *arXiv preprint arXiv:1605.09782* (2016).

[63] P. Bojanowski et al. “Optimizing the Latent Space of Generative Networks”. In: *stat* 1050 (2017), p. 18.

Chapter 6

Process-Specific Approaches for Vocal Fold Modeling

Process-specific models deal with physical processes and model them with dynamical systems. In the context of VFAH, we are interested in the physical (bio-mechanical) process of voice production, particularly phonation. By formulating process-specific models, we can study the dynamics of voice production within the phase space of these models and characterize various profiling parameters based on them.[1] We use these models to work with the specific case of identifying and characterizing different voice abnormalities.

6.1 Brief Introduction to Dynamical Systems

Our study primarily concerns real-time dynamical systems.

Definition 6.1 (Dynamical system)**.** A real-time dynamical system is a tuple (T, M, Φ), where T is a monoid (an algebraic construct, such as an open interval in $\mathbb{R}_+$). M is a manifold locally diffeomorphic to a Banach space, usually called the *phase space*. As

[1]Note here that we are not specifically interested in synthesizing speech with them—that would involve modeling other linguistic, prosodic, articulatory, and co-articulatory phenomena inherent in speech. Instead, we are simply interested in using them to model the process of phonation and deduce the physical and aerodynamic properties of the vocal folds, based on which we expect to be able to make accurate deductions about the underlying factors that influence the speaker's voice.

opposed to the configuration space describing the "position" of a dynamical system, the phase space describes the "states" or "motion" of the dynamical system. It is often defined as the tangent bundle TM or the cotangent bundle T^*M of the underlying manifold. $\Phi : T \times M \supseteq U \to M$, where $\text{proj}_2(U) = M$, is the (continuous) evolution function [1].

Definition 6.2 (Evolution function)**.** Denote the duration of evolution of a dynamical system as $I(x) = \{t \in T \mid (t, x) \in U\}$. The evolution function Φ is a group action of T on M satisfying

1. $\Phi(0, x) = x,$ for all $x \in M$;
2. $\Phi(t_2 + t_1, x) = \Phi(t_2, \Phi(t_1, x)),$ for $t_1, t_2 + t_1 \in I(x), t_2 \in I(\Phi(t_1, x))$.

A dynamical system can be instantiated with ordinary or partial differential equations with initial conditions, and the evolution function Φ is the solution to the ODE or PDE. We write $\Phi_x(t) \equiv \Phi^t(x) \equiv \Phi(t, x)$. The map $\Phi^t : M \to M$ is a diffeomorphism (i.e., differentiable, invertible, bijection map between manifolds).

Definition 6.3 (Flow, orbit, invariance)**.** The map $\Phi_x : I(x) \to M$ is the *flow* or *trajectory* through x. The set of all flows $\gamma_x \coloneqq \{\Phi_x \mid t \in I(x)\}$ is the *orbit* through x. Particularly, a subset $S \subseteq M$ is called Φ-*invariant* if $\Phi(t, x) \in S$ for all $x \in S$ and $t \in T$.

The behaviors of flows can be described by their attractor/attraction sets.

Definition 6.4 (Attractor)**.** An attractor set $A \subseteq M$ in the phase space is a closed subset satisfying the condition that for an initial point x, there exists a t_0 such that $\Phi_x(t) \in A$ for any $t > t_0$.

Namely, the orbit γ_x is "trapped" in the interior of A. A dynamical system can have more than one attractor set depending on the initial points (or the choice of parameters, as we will see later). Locally we can talk about a *basin of attraction* $B(A)$, which is a neighborhood of A satisfying for any initial point $x \in B(A)$, and its orbit is eventually trapped in A. There are different types of attractor sets, and the specific

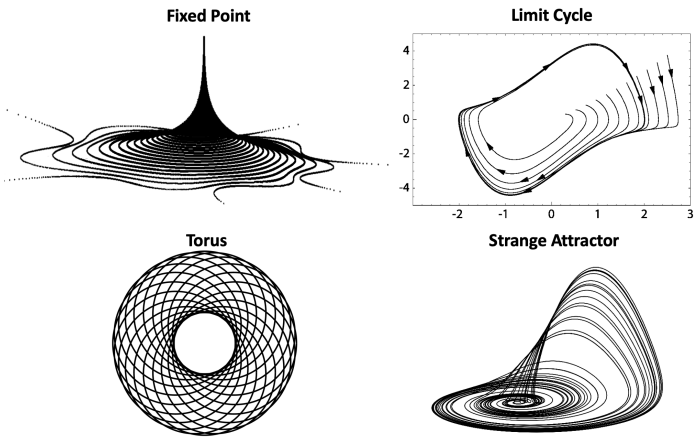


Figure 6-1: Illustration of different attractors in a dynamical system's phase space.

types exhibited by the system depend on its parameter settings. Some are shown in Figure 6-1. The simplest one is a fixed point or an equilibrium point, to which the solutions of the system eventually converge, regardless of the initial setting of the variables. We are particularly interested in those attractors that exhibit periodic motion of the flow (the solution trajectory) in phase space. Such attractors include the limit cycle or the limit torus, an isolated periodic or toroidal orbit. Some attractor sets have a fractal structure emerging from a chaotic state of the dynamical system [2, 3]. Chaos is a characteristic state of a nonlinear dynamic system. There are different definitions of chaos. Putting it simply

Definition 6.5 (Chaos)**.** Equip a distance metric d on the phase space M. Then $C \in M$ is referred to as a chaotic set of Φ if, for any $x, y \in C$, $x \neq y$, we have

$$\lim_{n\to\infty} \inf d(\Phi^n(x), \Phi^n(y)) = 0 \tag{6.1}$$

$$\lim_{n\to\infty} \sup d(\Phi^n(x), \Phi^n(y)) > 0 \tag{6.2}$$

This captures the system's sensitivity to initial conditions when it is in a state of chaos—for any two arbitrarily close initial points, the solution trajectories will diverge in phase space, and the rate of divergence is exponential. This characteristic of the exponential rate of divergence is captured by the *Lyapunov exponent*, which also measures the sensitivity of the evolution of the dynamical system to initial conditions.

These different types of attractor sets exhibit different levels of stability of dynamical systems.

Definition 6.6 (Stability)**.** A compact Φ-invariant subset $A = \Phi(A) \subseteq M$ is called a *Lyapunov stable* attraction set if

1. It has an open basin of attraction $B(A)$;
2. The Lyapunov stability condition is satisfied: every neighborhood U of A contains a smaller neighborhood V such that every iterative forward image $\Phi^n(V)$ is contained in U.

To study the orbit structure of the trajectories of dynamical systems, we use the Poincaré map or Poincaré section.

Definition 6.7 (Poincaré map [1])**.** For an n-dimensional phase space with a periodic orbit γ_x, a Poincaré section S is an $(n-1)$-dimensional section (hyper-plane) that is transverse to γ_x. Given an open, connected neighborhood $U \subseteq S$ of x, the Poincaré map on Poincaré section S is a map $P : U \to S$, $x \mapsto \Phi_x(t_s)$ where t_s is the time between the two intersections, satisfying

1. $P(U)$ is a neighborhood of x and $P : U \to P(U)$ is a diffeomorphism;
2. For every point x in U, the positive semi-orbit of x intersects S for the first time at $P(x)$.

Since the flow of a dynamical system in its phase space is a function of its parameters, the topological structure of the trajectories (including attractor sets) in phase space changes as the parameters change. To see how the topological structure changes with system parameters, we study the bifurcation map of the system.

Definition 6.8 (Bifurcation)**.** A bifurcation occurs when a small smooth change in a system parameter value causes an abrupt change in the topological structure of the trajectory in phase space. A *bifurcation diagram* is a visualization of the system's parameter space showing the number and behavior of attractor sets for each parameter configuration.

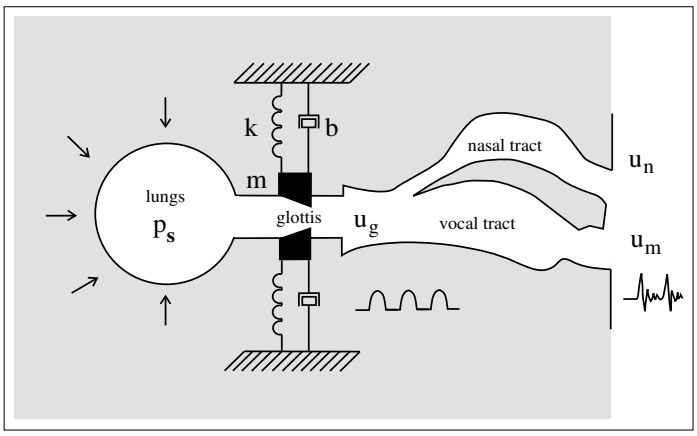


Figure 6-2: Illustration of the phonation process. Airflow from the lungs, driven by the subglottal pressure P_s, passes through the glottis, and vocal folds are set into a state of self-sustained vibration, producing the glottal flow u_g which is a quasi-periodic pressure wave. The vibration of vocal folds is analogous to a pair of mass-spring-damper oscillators. Further, the glottal flow resonates in the speaker's vocal tract and produces voiced sound.

At a *bifurcation point*, the system stability may change as the topological structure splits or merges, such as the periodic doubling or halving of a limit cycle.

6.2 Phonation Modeling and Characterization

Phonation is the process wherein the vocal folds in the larynx are set into a state of self-sustained vibration, causing an excitation signal to be produced at the glottal source. This signal, called the *glottal flow*, is a quasi-periodic pressure wave at a fundamental frequency (the pitch) of a few hundred hertz. Further, it resonates in the speaker's vocal tract, consisting of the laryngeal cavity, the pharynx, the oral cavity, and the nasal cavity. Depending on the vocal tract's shape and the articulators' configuration (tongue, lip, jaw, etc.), the pressure wave is heard as a characteristic voiced sound by the listener. Figure 6-2 illustrates the phonation process. Phonation is important in the production of all vowels and all voiced consonants in all languages of the world.

At the biomechanical level, phonation happens due to a specific pattern of events in the glottal region. The vocal folds are membranes that are set into self-sustained

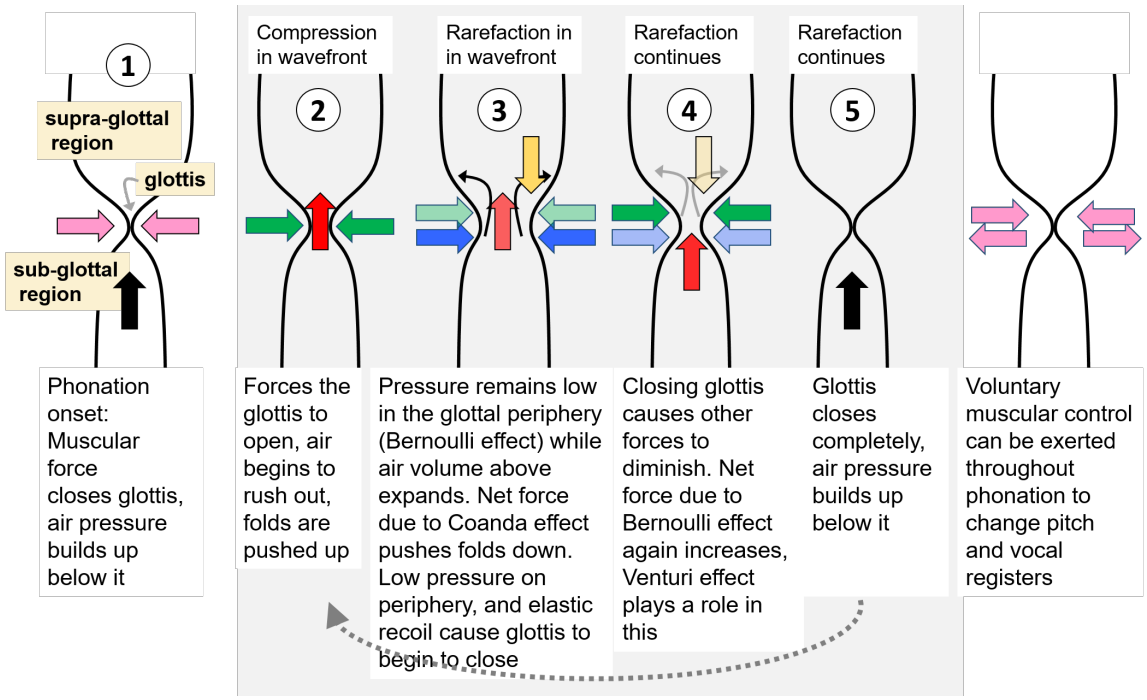


Figure 6-3: Schematic of the balance of forces through one cycle of the self-sustained vibrations of the vocal folds. The color codes for the arrows depict net forces due to the following: Pink–muscular; Green–Bernoulli effect; Yellow–Coandă effect; Blue–vocal fold elasticity and other factors; Black and Red–air pressure. Lighter shades of each color depict smaller forces. Figure from [6] with permission.

vibratory motion. By the myoelastic-aerodynamic theory of phonation, such motion is initiated and driven by a complex and delicate interplay of physical and aerodynamic forces in the laryngeal region [4, 5]. These forces relate to (a) pressure balances and airflow dynamics within the supra-glottal and sub-glottal regions and (b) muscular control within the glottis and the larynx. The balance of forces necessary to cause self-sustained vibrations is created by two physical phenomena: the Bernoulli effect and the Coandă effect. Figure 6-3 illustrates the interaction between these effects that drives the oscillations of the vocal folds.

The process of phonation begins with the closing of the glottis. This closure is voluntary and facilitated by the laryngeal muscles. Once closed, the muscles do not actively play a role in sustaining the vibrations. Glottal closure is followed by a contraction of the lungs which pushes out air and causes an increase in pressure just below the glottis. When this subglottal pressure crosses a threshold, the vocal folds

are pushed apart, and air rushes out of the narrow glottal opening into the much wider supra-glottal region, creating negative intra-glottal pressure (with reference to atmospheric air pressure) [6].

From the airflow perspective, the glottis thus forms a flow-separation plane. The air expansion in this region and the low pressure created in the vicinity of the glottis through the Coandă effect-induced entrainment cause a lowering of pressure close to the glottis and a net downward force on the glottis. At the same time, lowered pressure in the glottal region due to the Bernoulli effect that ensues from the high-velocity air volume flow through the glottis exerts a negative force on the glottis. The negative Bernoulli pressure causes elastic recoil, causing it to begin to close again. The closing reduces the volume flow through the glottis, diminishing the downward forces acting on it. Increased pressure buildup in the sub-glottal region causes the glottis to open again. This chain of oscillations continues in a self-sustained fashion throughout phonation until voluntary muscle control intervenes to alter or stop it or as the respiratory volume of air in the lungs is exhausted [6].

6.2.1 Phonation Models

Physical models of phonation attempt to explain this complex physical process using relations derived from actual physics, especially aerodynamics and the physics of mechanical structures. The exact physics of the airflow through the glottis during phonation is well studied, e.g., [5, 7, 8, 9, 10, 11], and several physical models have been proposed for it, e.g., [6, 8, 12, 13, 14, 15, 16, 17]. Specifically, the phonation process can be divided into two sub-processes: (1) vocal folds oscillation, and (2) wave propagation in the vocal tract. Correspondingly, the models include *vocal fold models* and *vocal tract models*. The vocal fold models describe the vibration of vocal folds and their aerodynamic interaction with airflow. Such models are of four broad types: one-mass models e.g. [5, 13, 18, 19, 20], two-mass models e.g. [8, 12], multi-mass models [15], and finite element models [14]. Each of these has proven to be useful in different contexts. On the other hand, the vocal tract models describe the interaction of the acoustic pressure wave with the vocal chambers. The vocal tract can be described by statistical

models [21], geometric models [22], biomechanical models [23], etc. In order to describe the aero-acoustic interaction of airflow and vocal tract, different models are applied, such as the reflection type line analog model, the transmission line circuit analog model [24], hybrid time-frequency domain models [25], finite-element models [26], etc.

One-mass model One-mass models describe vocal fold vibration as a mass-damper-spring oscillator. As an example

$$M\ddot{x} + B\dot{x} + Kx = f(x, \dot{x}, t)$$

where x is lateral displacement, M, B, K are mass, damping, and stiffness coefficients, f is the driving force, and t is time [5]. The driving force is velocity-dependent and can be estimated by Bernoulli's energy law

$$P_g = P_s - \frac{1}{2}\rho v^2$$

where P_g is mean glottal pressure, P_s is sub-glottal pressure, ρ is air density, and v is air-particle velocity. The kinetic pressure in the supra-glottal region is neglected [5].

Two-mass model Two-mass models describe vocal fold motion as two coupled mass-damper-spring oscillators. As an example

$$M_1\ddot{x}_1 + B_1\dot{x}_1 + K(x_1 - x_2) + R_1 = F_1$$
$$M_2\ddot{x}_2 + B_2\dot{x}_2 + K(x_2 - x_1) + R_2 = F_2$$

where x_i, M_i, and B_i are the i-th oscillator's displacement, mass, and viscous damping coefficient, respectively, K is the coupling stiffness between the two masses, F_i is the driving force, and R_i is the elastic restoring force [12]. This model assumes (1) small air inertia and quasi-steady glottal flow, (2) negligible supra-glottal pressure, and (3) that the nonlinearity induced by vocal fold collision is small. These assumptions lead to small-amplitude oscillations and allow model simplification [12].

Multi-mass model Multi-mass models have a large degree of freedom and hence can model vocal fold motion with high precision. For the i-th mass component, their equation of motion is

$$M_i \ddot{\boldsymbol{x}}_i = \boldsymbol{F}_i^A + \boldsymbol{F}_i^V + \boldsymbol{F}_i^L + \boldsymbol{F}_i^C + \boldsymbol{F}_i^D$$

where $\boldsymbol{x}_i = (x_i, y_i, z_i)$ is the three-dimensional displacement, M_i is the mass, $\boldsymbol{F}_i^A$ is the anchor force associated with the anchor spring and damping, $\boldsymbol{F}_i^V$ and $\boldsymbol{F}_i^L$ are the vertical and longitudinal coupling forces associated with spring and damping, $\boldsymbol{F}_i^C$ is the collision restoring force, and $\boldsymbol{F}_i^D$ is the driving force from glottal pressure [15]. In [15], 50 masses are used.

Finite element model Finite element models discretize the vocal fold motion in space and time—the geometry of the vocal fold is discretized into small elements (cells). In each cell, the differential equation governed by the laws of physics is solved. These models can handle complex geometries, continuous deformation, and complex driving forces [14]. Consider a cube element with six stress and strain components. By the principles of mechanics for elasticity-mediated movements

$$\boldsymbol{\sigma} = \boldsymbol{S}\boldsymbol{\epsilon}$$

where $\boldsymbol{\sigma}$ is the stress tensor, $\boldsymbol{\epsilon}$ is the strain tensor, and $\boldsymbol{S}$ is the stiffness matrix consisting of Young's modulus, shear modulus, and Poisson's ratio [14]. The relation

between stress and displacement is governed by

$$\begin{aligned}
\sigma_x &= C_1\mu\frac{\partial u}{\partial x} + C_2\mu\frac{\partial w}{\partial z} \\
\sigma_z &= C_2\mu\frac{\partial u}{\partial x} + C_1\mu\frac{\partial w}{\partial z} \\
\tau_{xy} &= \mu'\frac{\partial u}{\partial y} \\
\tau_{yz} &= \mu'\frac{\partial w}{\partial y} \\
\tau_{zx} &= \mu\left(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z}\right)
\end{aligned}$$

where τ is the shear stress, u and w are the lateral and vertical components of the displacement vector, μ and μ' are shear modulus, and C_1 and C_2 are constants [14]. This system of partial differential equations can be efficiently solved by finite element methods. We will describe this in more detail in the next chapter.

Vocal tract models in general The vocal tract models fall into one of three categories—statistical, geometrical, or biomechanical. **Statistical models** describe the vocal tract as statistical factors or components. For instance, factor analysis describes the vocal tract profile as a sum of articulatory components and analyzes the relationship between individual or combination of components and vocal tract parameters [21]. **Geometric models** attempt to depict the shape and geometric configurations of the vocal tract. They specify the articulatory state with vocal tract parameters that define the position and shape of the tongue, lips, jaw, larynx, etc [22]. However, such models are not scalable because they do not account for the continuous variations of the anatomy and articulatory state, require clinical measurements such as from magnetic resonance imaging, and are not amendable to coupling with vocal fold models. On the other hand, **biomechanical models** are more scalable and accurate. They simulate the geometry and articulatory movements of the vocal tract using displacement-based finite element methods and take into account the continuous tissue deformation and variation of the physiological, biomechanical, and viscoelastic properties of muscles [23]. Consequently, such models lend us more fine-grained control

over muscular forces, articulator positions, and movements. To study the interaction between vocal folds and the rest of the vocal tract, modeling approaches often take analog processes into the digital circuit regime and model the propagation of glottal flow in the vocal tract as a transmission line circuit [24]. One can evaluate the system (vocal tract)'s transfer function in the time and frequency domain and acquire the system output in response to the input (glottal flow) [25]. We take a different approach by uniting the vocal folds and tract into a single model. We efficiently solve the vocal fold-tract model and estimate model parameters directly from recorded speech measurements. We dedicate the next chapter to studying such integration of the vocal fold and tract model.

Asymmetric one-mass body-cover model for disordered phonation In this study, we are interested in process-specific modeling and characterization of disordered phonations as an illustrative example. This serves the additional purpose of giving us tools to aid the diagnosis and treatment of voice pathologies. The term "voice disorders" refers to any abnormality wherein voice quality differs from its normal status [27]. The abnormality can be physiological, i.e., due to the structural alteration of voice apparatus, such as edema or occurrence of vocal nodules, or neurogenic changes, such as vocal tremor, spasmodic dysphonia, or paralysis of vocal folds. The abnormality can also be functional, i.e., due to the improper use of the voice production apparatus, such as vocal fatigue, muscle tension dysphonia, aphonia, diplophonia, or ventricular phonation.

In most vocal pathologies such as vocal palsy, phonotrauma, neoplasm, etc., the properties of the vocal structures vary from their generic settings [28]. These often cause the movements of the vocal folds to become asymmetric [16, 29]—where the movements of the left and right folds are out of sync—in a manner that is characteristic of the underlying pathology. For our purpose, the one-mass asymmetric body-cover model [5, 18, 19, 20] is adopted, as illustrated in Figure 6-4. This model incorporates an asymmetry parameter, which can emulate the asymmetry in the vibratory motions of the left and right vocal folds. Hence, it is ideally suited to modeling pathological

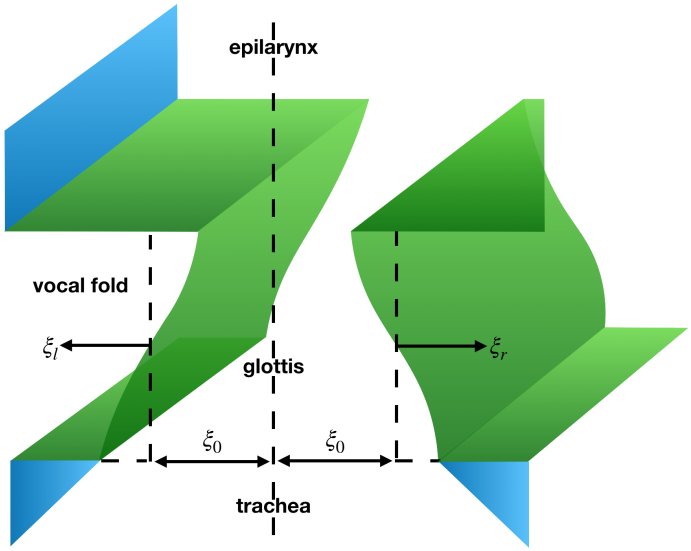


Figure 6-4: Diagram of the one-mass body-cover model for vocal folds. The lateral displacements at the midpoint of the left and right vocal folds are denoted as ξ_l and ξ_r, and ξ_0 represents the half glottal width at rest.

phonation [30]. The key assumptions made in formulating this model are:

(a) The degree of asymmetry is independent of the oscillation frequency;

(b) The glottal flow is stationary, frictionless, and incompressible;

(c) All subglottal and supraglottal loads are neglected, eliminating the effect of source-vocal tract interaction;

(d) There is no glottal closure and hence no vocal fold collision during the oscillation cycle;

(e) The small-amplitude body-cover assumption is applicable (see definition below).

Assumption 6.1 (Body-cover assumption)**.** The body-cover assumption assumes that a glottal flow-induced mucosal wave travels upwards within the transglottal region, causing a small displacement of the mucosal tissue, which attenuates down within a few millimeters into the tissue as an energy exchange happens between the airstream and the tissue [5].

This assumption allows us to represent the mucosal wave as a one-dimensional surface wave on the mucosal surface (the cover) and treat the remainder of the vocal folds

(the body) as a single mass or safely neglect it. As a result, the oscillation model can be linearized, and the oscillatory conditions are much simplified while maintaining the model's accuracy. We adopt the specific formulation of the one-mass asymmetric model from [20]. As shown in Figure 6-4, the left and right vocal folds oscillate with lateral displacement ξ_l and ξ_r, resulting in a pair of coupled Van der Pol oscillators

$$\ddot{\xi}_r + \beta(1+\xi_r^2)\dot{\xi}_r + \xi_r - \frac{\Delta}{2}\xi_r = \alpha(\dot{\xi}_r + \dot{\xi}_l)$$
$$\ddot{\xi}_l + \beta(1+\xi_l^2)\dot{\xi}_l + \xi_l + \frac{\Delta}{2}\xi_l = \alpha(\dot{\xi}_r + \dot{\xi}_l)$$

where β is the coefficient incorporating mass, spring, and damping coefficients, α is the glottal pressure coupling coefficient, and Δ is the asymmetry coefficient.

Inverse problem of model parameter estimation In the case of both the vocal folds model and vocal tract model—their actual dynamics, i.e., the flows in phase space, are governed by various biomechanical parameters of the vocal folds such as elasticity, resistance, Young's modulus, viscosity, etc., as well as the configurations of vocal tract such as time-varying cross-sectional area. While these models effectively solve the *forward* problem of accurately emulating vocal fold and vocal tract dynamics during phonation, the *inverse* problem of finding the correct model parameters given a set of observed speech signals has not been addressed. Hence, our research problem is: (1) how can we effectively solve the inverse problem of accurately estimating the parameters of vocal folds and vocal tract models (and hence their dynamics) from observed voice signals, and (2) how can we use the model dynamics to characterize pathological phonation?

The inverse problem is challenging to solve in real-life settings. For example, to estimate the parameters for the vocal folds oscillation model, one needs to consider the vocal tract coupling, the effect of lossy medium and lip radiation, etc. We eventually find this problem intractable as we add more factors to consider. Two schools of approaches have been proposed to ease the difficulty induced by vocal folds–tract coupling. One is to isolate and only examine the vocal folds model. For this, however, one must acquire

measurements of the vocal fold displacements. This in turn requires either high-speed photography [31] or physical or numerical simulations [14, 32], which are often not easily accessible. Even with the measurements, solving the inverse problem remains hard [33]. It is usually solved via iterative matching procedures [34, 35, 36], stochastic optimization, or heuristic procedures [15, 37]. Alternative approaches attempt to discretize the vocal tract with consecutive, cross-sectional area varying tubes or with a mesh-grid [38, 39], simplifying the solution. However, such approximation increases the estimation error.

Forward and backward approaches for inverse problems To address the problems inherent in conventional approaches, we propose a solution incorporating a backward approach and a forward approach. The backward approach eliminates the need for a vocal tract model by estimating the glottal flow from speech signals via inverse filtering. As a specific instance of this approach, we propose an adjoint least-squares (ADLES) method [40] to effectively solve an ODE-constrained functional minimization problem and hence, accurately estimate the parameters of the asymmetric vocal folds model.

On the other hand, the forward approach combines the vocal folds oscillation model and the vocal tract propagation model. In the simplest case, the vocal folds oscillation model is a one-mass model with asymmetry parameters described by coupled ODEs. The vocal tract model is an acoustic wave propagation model described by PDEs. Combined, they accurately represent phonation for both normal and disordered voices. As an instance of this approach, we propose an iterative adjoint method to solve the ODE/PDE constrained inverse problem. It enables the estimation of model parameters directly from speech measurements. Our approach significantly alleviates the difficulty of obtaining physical measurements in clinical settings while at the same time promoting model accurateness.

Once we recover the model parameters through our backward or forward approach, we further show how the re-estimated model parameters can be mapped into the phase space of the nonlinear dynamical systems and how the location of these parameters

in the model parameter space can directly indicate the underlying pathology in the observed speech signal. Since the vocal fold dynamics are nonlinear, the models are systems of coupled nonlinear dynamical equations. They output a phase space trajectory of state variables that describes the movements of the vocal folds. The trajectories tend to fall into orbits with regular or irregular behaviors that explain observed behavior patterns of the vocal folds. The possible types and distributions of these orbits depend on the system parameters.

Having broadly introduced the approaches and tools for modeling and characterizing normal and pathological phonation, next, we present a specific example. Taking the backward approach, in the following sections, we propose an efficient method to estimate the parameters of the asymmetric vocal fold oscillation model and use them to classify vocal fold pathologies. The next chapter will extend this and present the forward approach.

6.3 Speech-Based Parameter Estimation of a Vocal Fold Model for Voice Pathology Discrimination

So far, several physical models have been proposed to study vocal fold oscillations during phonation. The parameters of these models, such as vocal fold elasticity, resistance, etc., are traditionally determined through observing and measuring the vocal fold vibrations in the larynx. Since such direct measurements tend to be the most accurate, the traditional practice has been to set the parameter values of these models based on averaged measurements across an ensemble of human subjects. However, the direct measurement process is hard to revise outside clinical settings. In many cases, especially in pathological ones, the properties of the vocal folds often deviate from their generic values—sometimes asymmetrically, wherein the characteristics of the two vocal folds differ for the same individual. In such cases, it is desirable to find a more scalable way to adjust the model parameters on a case-by-case basis.

We present a novel and alternate way to solve the inverse problem of phonation:

determining vocal fold model parameters from the speech signal. Given a model for asymmetric movements of the vocal folds and a set of speech signals from people affected by various pathologies (which affect vocal fold movements), we propose a method to estimate the parameters of the asymmetric model that explains them. We further show that for such models, differences in estimated parameters can be successfully used to discriminate between voices characteristic of different underlying vocal fold pathologies.

The premise of this study is that if these movements of the vocal folds and the underlying parameters of the system that produces them could also be recovered from the speech signal, the underlying pathologies could be identified. Note that this diverges from the traditional approach of classifying these through analysis of the surface-level waveform. In contrast, we present this novel paradigm wherein the goal is to estimate the actual vocal fold dynamics from the waveform. In order to do so, we must consider the actual physics of the vocal-fold movements, the physical properties of the vocal folds, how they influence their movements, and how these manifest in the speech signal itself.

6.3.1 The Asymmetric Vocal Folds Oscillation Model

For this study, we consider the asymmetric one-mass body-cover model [5, 18, 19, 20] as described in the previous section. The vibration of vocal folds is modeled with a pair of mass-damper-spring oscillators, as shown in Figure 6-2 and 6-4. Adopting the specific formulation in [20], we denote the center-line of the glottis as the z-axis. At the midpoint ($z = 0$) of the thickness of the vocal folds, the left and right vocal folds oscillate with lateral displacement ξ_l and ξ_r, resulting in a pair of coupled Van der Pol oscillators

$$\begin{aligned}
\ddot{\xi}_r + \beta(1+\xi_r^2)\dot{\xi}_r + \xi_r - \frac{\Delta}{2}\xi_r &= \alpha(\dot{\xi}_r + \dot{\xi}_l) \\
\ddot{\xi}_l + \beta(1+\xi_l^2)\dot{\xi}_l + \xi_l + \frac{\Delta}{2}\xi_l &= \alpha(\dot{\xi}_r + \dot{\xi}_l)
\end{aligned} \tag{6.3}$$

where β is the coefficient incorporating mass, spring, and damping coefficients, α is the glottal pressure coupling coefficient, and Δ is the asymmetry coefficient. For a male adult with normal voice, the reference values for the model parameters may be $\alpha = 0.5$, $\beta = 0.32$ and $\Delta = 0$.

6.3.2 Physical Interpretation of Phase Space of Asymmetric Model

We have introduced the concepts and tools used to study the behaviors (e.g., flow, orbit, attractor, stability, Poincaré map, bifurcation) of nonlinear dynamical systems such as (6.3) in the previous sections. The phase space of the system in (6.3) is four-dimensional and includes states $(\xi_r, \dot{\xi}_r, \xi_l, \dot{\xi}_l)$. For this nonlinear system, it is expected that attractors such as limit cycles or toruses will appear in the phase space. Such phenomena are consequences of specific parameter settings. Specifically, the parameter β determines the periodicity of oscillations; the parameter α and Δ quantify the asymmetry of the displacement of left and right vocal folds and the degree to which one of the vocal folds is out of phase with the other [20, 29]. We can visualize them by plotting the left and right displacements and the phase space portrait.

The coupling of right and left oscillators is described by their *entrainment*; they are in $n : m$ entrainment if their phase θ_r, θ_l satisfy $|n\theta_r - m\theta_l| < C$ where n, m are integers and C is a constant [20]. Such entrainment can be shown by the Poincaré map, where the number of trajectory crossings of the right or left oscillator with the Poincaré section shows the periodicity of its limit cycles. Therefore, their ratio represents the entrainment. We can use the bifurcation diagram to visualize how the entrainment changes with parameters. An example of such a bifurcation diagram is shown in Figure 6-5 [12, 29]. As we will see later (and as indicated in Figure 6-5), model parameters can characterize voice pathologies, which will also be visible in phase portraits and bifurcation plots.

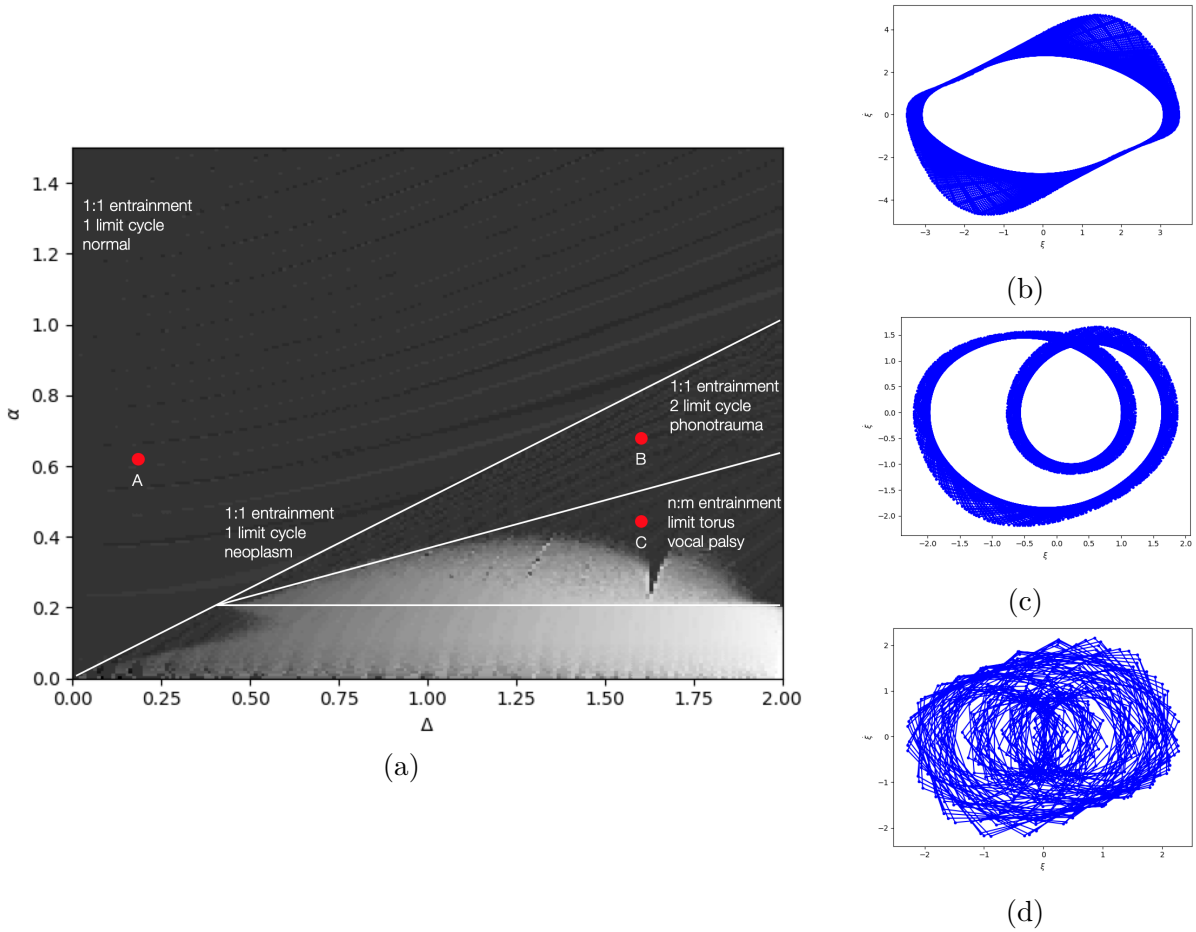


Figure 6-5: Bifurcation diagram of the asymmetric vocal fold model. It shows the entrainment ratio $n : m$ (coded as different shades of grey) as a function of model parameters α and Δ, where n and m are the number of intersections of the orbits of right and left oscillators across the Poincaré section $\dot{\xi}_{r,l} = 0$ at stable status. This is consistent with the theoretical results in [20]. **(b)**, **(c)**, and **(d)** show the phase portraits for points A, B, and C, where the horizontal axis is displacement and the vertical axis is velocity.

6.3.3 Model Parameter Estimation

Finding the parameters of any physical model that emulates vocal fold oscillations is not trivial. For this, one must acquire measurements of the vocal fold displacements, which in turn require either high-speed photography [31] or physical or numerical simulations [14, 32], which are often not easily accessible. Even with the measurements, estimating the model parameters remain hard. The problem itself is commonly termed as the *inverse problem* [33], and is usually solved via iterative matching procedures [34,

35, 36], stochastic optimization or heuristic procedures [15, 37].

We propose a method to solve the inverse problem and bypass the difficulties inherent in traditional methods, namely that of either obtaining direct measurements of vocal fold displacements or of the complexity of solving inverse problems. Our proposed solution comprises an adjoint least-squares (ADLES) method [40] to effectively solve an ODE-constrained functional minimization problem and hence, accurately estimate the parameters of the asymmetric vocal folds model directly from speech measurements.

First, we formulate our objective. The vibration of vocal folds oscillates the air particles at the glottal region, producing a pressure wave that propagates through the upper vocal channel into the open air. This pressure wave is considered planar when its frequency is under 4 kHz [41], and hence a function of position $x \in \Omega$ and time $t \in T$: $p(x,t) \in \mathcal{L}^2(\Omega \times T)$, where $\Omega \coloneqq [0, L]$, L is the length of the upper vocal channel, and $T \coloneqq [0, t_m]$ for maximum duration t_m. The acoustic pressure $p_L(t) \coloneqq p(L,t)$, which represents the speech signal measured by a microphone near the mouth, is a result of the pressure wave $p_0(t) \coloneqq p(0,t)$ at the glottis modulated by the upper vocal channel. If we denote the effect of the upper vocal channel as a filter

$$\mathcal{F} : \mathcal{L}^2(T) \to \mathcal{L}^2(T) \tag{6.4}$$

$$p_0(t) \mapsto p_L(t) \tag{6.5}$$

we can deduce $p_0(t)$ from $p_L(t)$ using inverse filtering [42]

$$p_0(t) = \mathcal{F}^{-1}(p_L(t)) \tag{6.6}$$

Let $A(x)$ be the area function of the vocal channel for $x \in [0, L]$ and $A(0)$ represent the cross-sectional area at the glottis. The corresponding volume velocity $u_0(t)$ through the vocal channel is given by

$$u_0(t) = \frac{A(0)}{\rho c} p_0(t) \tag{6.7}$$

where c is the speed of sound, and ρ is the ambient air density. As a result, given a

measured speech signal $p_m(t)$, we have

$$u_0^m(t) = \frac{A(0)}{\rho c}\mathcal{F}^{-1}(p_m(t)) \tag{6.8}$$

Alternatively, we can derive $u_0(t)$ from the displacement of vocal folds by

$$u_0(t) = \tilde{c}d\left(2\xi_0 + \xi_l(t) + \xi_r(t)\right) \tag{6.9}$$

where ξ_0 is the half glottal width at rest and is set to 0.1 cm, d is the length of vocal fold and is set to 1.75 cm, and $\tilde{c}$ is the air particle velocity at the midpoint of the vocal fold. Our objective is then to minimize the difference

$$\min \int_0^T \left(u_0(t) - u_0^m(t)\right)^2 dt \Leftrightarrow \tag{6.10}$$

$$\min \int_0^T \left(\tilde{c}d\left(2\xi_0 + \xi_l(t) + \xi_r(t)\right) - \frac{A(0)}{\rho c}\mathcal{F}^{-1}(p_m(t))\right)^2 dt \tag{6.11}$$

such that

$$\ddot{\xi}_r + \beta(1+\xi_r^2)\dot{\xi}_r + \xi_r - \frac{\Delta}{2}\xi_r = \alpha(\dot{\xi}_r + \dot{\xi}_l) \tag{6.12}$$

$$\ddot{\xi}_l + \beta(1+\xi_l^2)\dot{\xi}_l + \xi_l + \frac{\Delta}{2}\xi_l = \alpha(\dot{\xi}_r + \dot{\xi}_l) \tag{6.13}$$

$$\xi_r(0) = C_r \tag{6.14}$$

$$\xi_l(0) = C_l \tag{6.15}$$

$$\dot{\xi}_r(0) = 0 \tag{6.16}$$

$$\dot{\xi}_l(0) = 0 \tag{6.17}$$

where C_r and C_l are constants.

The Adjoint Least Squares Solution

To solve the functional least squares in (6.11), we require the gradients of (6.11) w.r.t. the model parameters α, β and Δ. Subsequently, we can adopt any gradient-based (local or global) method to obtain the solution. Denote the residual as $R =$

$\tilde{c}d\,(2\xi_0 + \xi_l(t) + \xi_r(t)) - \frac{A(0)}{\rho c}\mathcal{F}^{-1}(p_m(t))$; then $f(\xi_l, \xi_r; \vartheta) = R^2$, and $F(\xi_l, \xi_r; \vartheta) = \int_0^T f(\xi_l, \xi_r; \vartheta)dt$, where $\vartheta = [\alpha, \beta, \Delta]$. We define the Lagrangian

$$\begin{aligned}\mathcal{L}(\vartheta) = \int_0^T \Bigg[& f + \lambda \left(\ddot{\xi}_r + \beta \left(1 + \xi_r^2\right) \dot{\xi}_r + \xi_r - \frac{\Delta}{2}\xi_r - \alpha \left(\dot{\xi}_r + \dot{\xi}_l \right) \right) \\ & + \eta \left(\ddot{\xi}_l + \beta \left(1 + \xi_l^2\right) \dot{\xi}_l + \xi_l + \frac{\Delta}{2}\xi_l - \alpha \left(\dot{\xi}_r + \dot{\xi}_l \right) \right) \Bigg] dt \\ & + \mu_l \left(\xi_l(0) - C_l\right) + \mu_r \left(\xi_r(0) - C_r\right) + \nu_l \dot{\xi}_l(0) + \nu_r \dot{\xi}_r(0) \end{aligned} \tag{6.18}$$

where λ, η, μ and ν are Lagrangian multipliers. Taking the derivative of the Lagrangian w.r.t. the model parameter α yields

$$\begin{aligned}\mathcal{L}_\alpha = \int_0^T \Big[& 2\tilde{c}dR(\partial_\alpha \xi_l + \partial_\alpha \xi_r) \\ & + \lambda \Big(\partial_\alpha \ddot{\xi}_r + 2\beta \dot{\xi}_r \xi_r \partial_\alpha \xi_r + \beta \left(1 + \xi_r^2\right) \partial_\alpha \dot{\xi}_r \\ & + \partial_\alpha \xi_r - \frac{\Delta}{2}\partial_\alpha \xi_r - \alpha \left(\partial_\alpha \dot{\xi}_r + \partial_\alpha \dot{\xi}_l \right) - \left(\dot{\xi}_r + \dot{\xi}_r \right) \Big) \\ & + \eta \left(\partial_\alpha \ddot{\xi}_l + 2\beta \dot{\xi}_l \xi_l \partial_\alpha \xi_l + \beta \left(1 + \xi_l^2\right) \partial_\alpha \dot{\xi}_l \right. \\ & \left. + \partial_\alpha \xi_l + \frac{\Delta}{2}\partial_\alpha \xi_l - \alpha \left(\partial_\alpha \dot{\xi}_r + \partial_\alpha \dot{\xi}_l \right) - \left(\dot{\xi}_r + \dot{\xi}_r \right) \right) \Big] dt \\ & + \mu_l \partial_\alpha \xi_l(0) + \mu_r \partial_\alpha \xi_r(0) + \nu_l \partial_\alpha \dot{\xi}_l(0) + \nu_r \partial_\alpha \dot{\xi}_r(0) \end{aligned} \tag{6.19}$$

Integrating the term $\lambda \partial_\alpha \ddot{\xi}_r$ twice by parts yields

$$\int_0^T \lambda \partial_\alpha \ddot{\xi}_r dt = \int_0^T \partial_\alpha \xi_r \ddot{\lambda} dt - \partial_\alpha \xi_r \dot{\lambda} \Big|_0^T + \partial_\alpha \dot{\xi}_r \lambda \Big|_0^T \tag{6.20}$$

Applying the same to $\eta\partial_\alpha\ddot{\xi}_l$, substituting into (6.19) and simplifying the final expression we obtain

$$\begin{aligned}
\mathcal{L}_\alpha = \int_0^T \Bigg[& \left(\ddot{\lambda} + \left(2\beta\xi_r\dot{\xi}_r + 1 - \frac{\Delta}{2} \right) \lambda + 2\tilde{c}dR \right) \partial_\alpha \xi_r \\
& + \left(\ddot{\eta} + \left(2\beta\xi_l\dot{\xi}_l + 1 + \frac{\Delta}{2} \right) \lambda + 2\tilde{c}dR \right) \partial_\alpha \xi_l \\
& + \left(\beta(1+\xi_r^2)\lambda - \alpha(\lambda+\eta) \right) \partial_\alpha \dot{\xi}_r \\
& + \left((\beta(1+\xi_l^2)\eta - \alpha(\lambda+\eta) \right) \partial_\alpha \dot{\xi}_l \\
& - (\dot{\xi}_r + \dot{\xi}_l)(\lambda+\eta) \Bigg] dt \\
& + \left(\mu_r + \dot{\lambda} \right) \partial_\alpha \xi_r(0) - \dot{\lambda} \partial_\alpha \xi_r(T) \\
& + (\nu_r - \lambda)\, \partial_\alpha \dot{\xi}_r(0) + \lambda \partial_\alpha \dot{\xi}_r(T) \\
& + (\mu_l + \dot{\eta})\, \partial_\alpha \xi_l(0) - \dot{\eta} \partial_\alpha \xi_l(T) \\
& + (\nu_l - \eta)\, \partial_\alpha \dot{\xi}_l(0) + \eta \partial_\alpha \dot{\xi}_l(T)
\end{aligned} \tag{6.21}$$

Since the partial derivative of the model output ξ w.r.t. the model parameter α is difficult to compute, we eliminate the related terms by setting

For $0 < t < T$:

$$\ddot{\lambda} + \left(2\beta\xi_r\dot{\xi}_r + 1 - \frac{\Delta}{2} \right) \lambda + 2\tilde{c}dR = 0 \tag{6.22}$$

$$\ddot{\eta} + \left(2\beta\xi_l\dot{\xi}_l + 1 + \frac{\Delta}{2} \right) \eta + 2\tilde{c}dR = 0 \tag{6.23}$$

$$\beta \left(1 + \xi_r^2 \right) \lambda - \alpha \left(\lambda + \eta \right) = 0 \tag{6.24}$$

$$\beta \left(1 + \xi_l^2 \right) \eta - \alpha \left(\lambda + \eta \right) = 0 \tag{6.25}$$

with initial conditions

$$\text{At } t = T:$$

$$\lambda(T) = 0 \tag{6.26}$$

$$\dot{\lambda}(T) = 0 \tag{6.27}$$

$$\eta(T) = 0 \tag{6.28}$$

$$\dot{\eta}(T) = 0 \tag{6.29}$$

As a result, we obtain the derivative of F w.r.t. α as

$$F_\alpha = \int_0^T -\left(\dot{\xi}_r + \dot{\xi}_l\right)(\lambda + \eta)dt \tag{6.30}$$

The derivatives of F w.r.t. β and Δ are similarly obtained as

$$F_\beta = \int_0^T \left(\left(1 + \xi_r^2\right)\dot{\xi}_r\lambda + \left(1 + \xi_l^2\right)\dot{\xi}_l\eta\right)dt \tag{6.31}$$

$$F_\Delta = \int_0^T \tfrac{1}{2}\Big(\xi_l\eta - \xi_r\lambda\Big)dt \tag{6.32}$$

Having calculated the gradients of F w.r.t. the model parameters, we can now apply gradient-based algorithms to optimize our objective (6.11). For instance, applying gradient descent, we have

$$\begin{aligned}
\alpha^{k+1} &= \alpha^k - \tau^\alpha F_\alpha \\
\beta^{k+1} &= \beta^k - \tau^\beta F_\beta \\
\Delta^{k+1} &= \Delta^k - \tau^\Delta F_\Delta
\end{aligned} \tag{6.33}$$

where $\tau^{\cdot}$ is the step-size. The overall algorithm is summarized as follows:

1. Integrate (6.12) and (6.13) with initial conditions (6.14), (6.15), (6.16) and (6.17) from 0 to T, obtaining ξ_r, ξ_l, $\dot{\xi}_r$ and $\dot{\xi}_l$.

2. Integrate (6.22), (6.23), (6.24) and (6.25) with the initial conditions (6.26), (6.27), (6.28) and (6.29) from T to 0, obtaining λ, $\dot{\lambda}$, η and $\dot{\eta}$.

3. Update α, β and Δ with (6.33).

6.3.4 Experiments

We show the validity of our proposed ADLES method by using it to estimate the asymmetric model parameters for clinically acquired pathological speech data. We show that the estimated parameters can be effectively used to characterize the vocal disorders represented in our experimental data.

The data set used in our experiments is the FEMH dataset [28]. It has 200 voice samples of the sustained vowel sound `/a:/` obtained from a voice clinic in a tertiary teaching hospital, including 50 normal voice samples and 150 samples of common voice disorders. Within the disordered samples, there are 40/60/50 samples for glottis neoplasm, phonotrauma (including vocal nodules, polyps, and cysts), and unilateral vocal paralysis, respectively.

Figure 6-6 shows the glottal flows obtained by inverse filtering and those obtained by the asymmetric model with the parameters estimated by our ADLES method. We observe consistent matches, showing the accurateness of our estimations. Figure 6-7 shows phase portraits of the right and left vocal folds obtained with our ADLES method. We observe typical attractor behaviors for different types of pathologies. Table 6.1 shows the results of deducing voice pathologies by simple thresholding of parameter ranges. It validates that our ADLES method can accurately estimate model parameters and phase space behaviors and further use them to classify voice pathologies. Specifically, the ranges of model parameters in each row of Table 6.1 correspond to regions in the bifurcation diagram in Figure 6-5. Each region has distinctive attractors and phase entrainment, representing distinct vocal fold behaviors—thereby indicating different voice pathologies. By extracting the phase trajectories for the speech signal and, thereby, the underlying system parameters, the ADLES algorithm can be used to place the vocal-fold oscillations during phonation on the bifurcation diagram and thus deduce the underlying pathology.

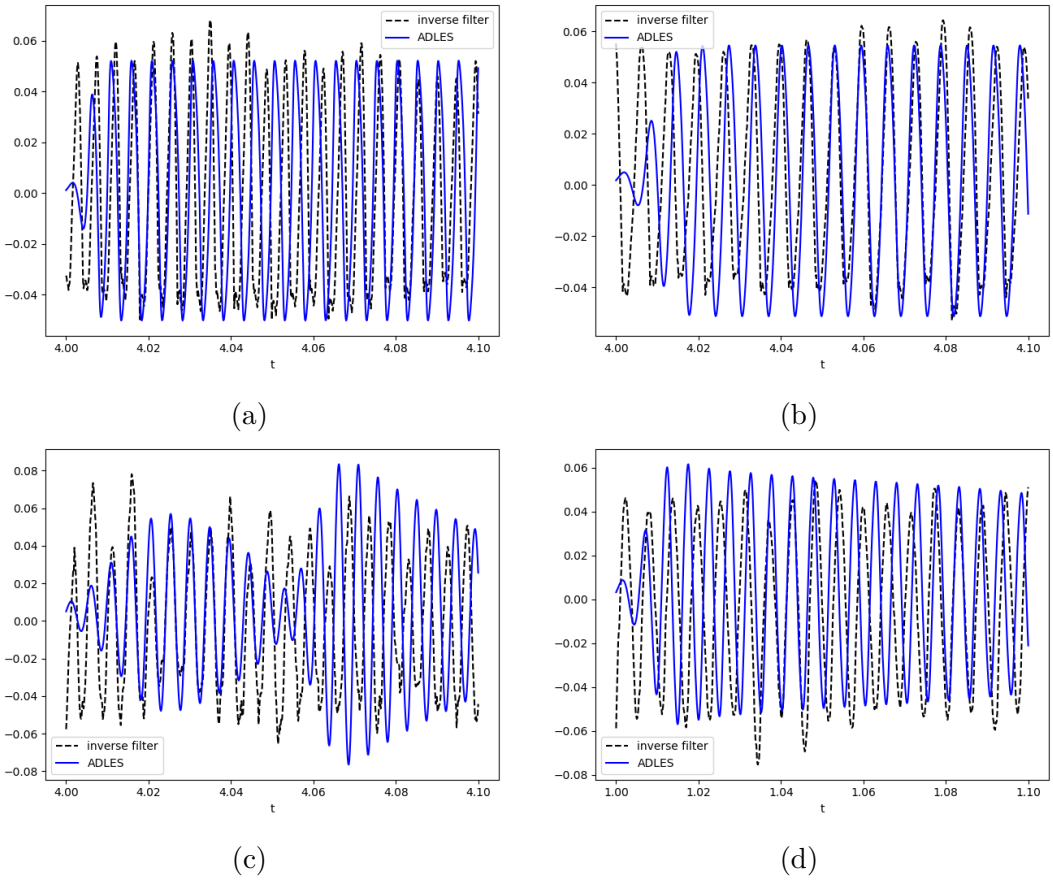


Figure 6-6: Glottal flows from inverse filtering and our ADLES estimation for **(a)** normal speech, **(b)** neoplasm, **(c)** phonotrauma, **(d)** vocal palsy.

Δ	α	Phase Space Behavior	Pathology	Accuracy
< 0.5	> 0.25	1 limit cycle, $1:1$ entrain	Normal	0.90
~ 0.6	~ 0.35	1 limit cycle, $1:1$ entrain	Neoplasm	0.82
~ 0.6	~ 0.3	2 limit cycles, $1:1$ entrain	Phonotrauma	0.95
~ 0.85	~ 0.4	toroidal, $n:m$ entrain	Vocal Palsy	0.89

Table 6.1: Parameters obtained and pathologies identified through ADLES.

6.4 Uniting Dynamical Systems with Machine Learning

We have presented a dynamical system approach for modeling physical processes. Next, we explore ways of uniting dynamical system modeling with machine learning approaches. We do this in two ways, noting many other possibilities. Firstly, we derive

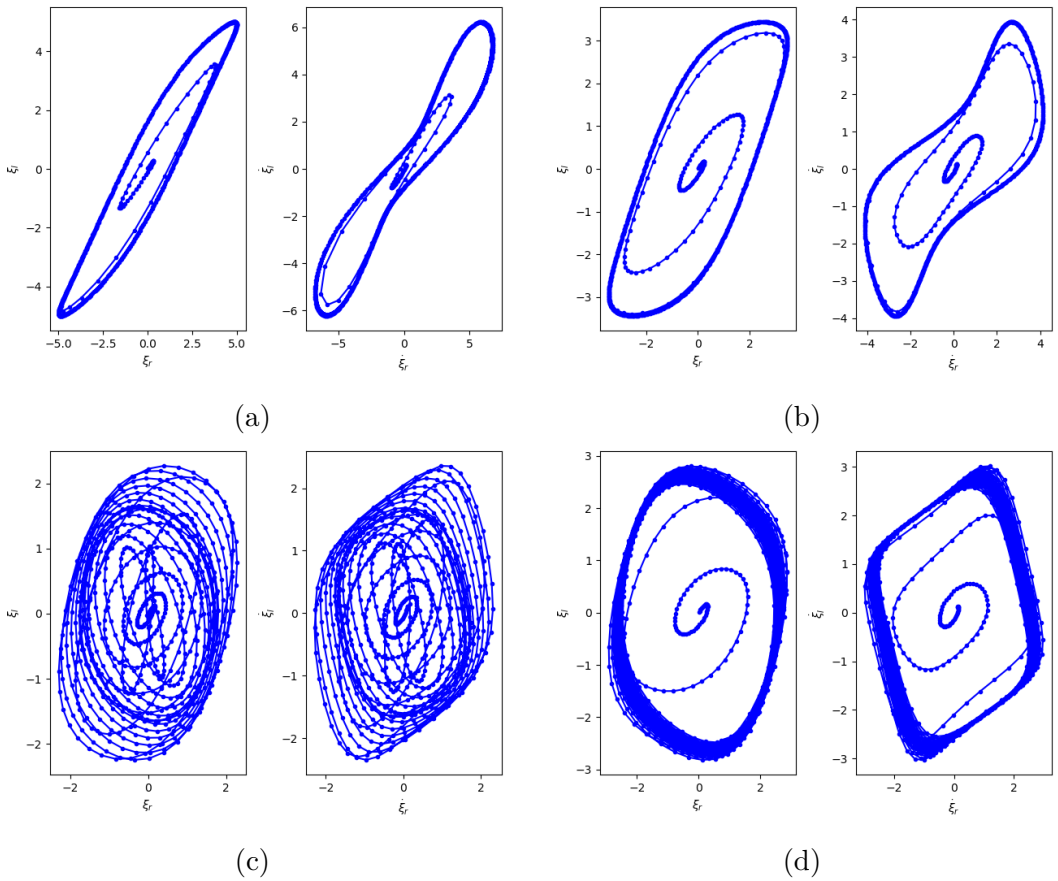


Figure 6-7: Phase portraits of left and right oscillators from our ADLES estimation for **(a)** normal speech–1 limit cycle, **(b)** neoplasm–1 limit cycle, **(c)** phonotrauma–2 limit cycles, **(d)** vocal palsy–limit torus.

features from dynamical systems that can be used by machine and deep learning models. In the second, we utilize the deep connection between neural models and dynamical systems.

6.4.1 Deriving Features From Dynamical Systems for Machine Learning

As a direct extension to our ADLES method that estimates model parameters for phonation processes, features can be derived from the phonation modeling to aid machine learning algorithms for specific voice-based detection tasks. In [43], the authors hypothesize that since COVID-19 would impair the human respiratory system, this, in turn, would affect the delicate phonation process and manifest in the vibration

signatures of vocal folds. They use the ADLES method to estimate the asymmetric vocal folds model parameters and the estimation residual and use the estimated model parameters as features to other binary classifiers such as logistic regression, support vector machine, decision tree, and random forest. They achieve 0.8 ROC-AUC (area under the ROC curve) for discriminating positive COVID-19 cases from clinically collected data of extended vowel sounds. The authors also discover that COVID-19 positive individuals display different phase space behaviors as compared to negative individuals: the phase space trajectories for negative individuals are more regular, while the trajectories for positive patients are more chaotic, implying a lack of synchronization and a higher degree of asymmetry in the vibrations of the left and right vocal folds. Further, authors in [44] use the ADLES-estimated glottal flows as features to CNN-based two-step attention neural networks. The neural model detects differences in the estimated and actual glottal flows and predicts two classes corresponding to COVID-19 positive and negative cases. It achieves 0.9 ROC-AUC on clinically collected vowel sounds. Another study uses higher-order statistics derived from parameters and the Lyapunov and Hurst exponents derived from the phase space trajectories of the asymmetric vocal folds model to detect Amyotrophic Lateral Sclerosis (ALS) from voice. It achieves high accuracy with normalized ROC-AUC of 0.82 to 0.99 [45].

6.4.2 Neural Models and Dynamical Systems

Identifying dynamical systems with neural models Neural models can be used for identifying dynamical systems and the underlying physical laws [46]. Neural nets can approximate ODE solutions with good precision, such as the finite neural element method proposed in [47]. Neural models infer the physical systems and governing laws of physics through data-driven approaches. Neural nets are learned from data sampled in the domain of the physical system, which is more efficient than finite-element methods due to mesh-free sampling. Neural nets integrated with dynamical systems can also make physically plausible forecasts [46]. Random processes are linked to stochastic differential equations and can be combined with neural models to deduce

system parameters from noisy observations [48]. For nonlinear dynamical systems, more powerful and advanced deep neural approaches such as convolutional neural nets, recurrent neural nets, encoder-decoder networks, and reinforcement learning can aid such inference [48].

Dynamical system perspective for neural modeling On the other hand, dynamical systems also shed light on neural models. Authors in [49] propose a NeuralODE model, formulated as a continuous-depth deep residual neural net and naturally extends to modeling continuous-time dynamics. The time-dependent hidden states in residual networks and recurrent neural nets are discrete and are made continuous by letting the time step go infinitesimal. This essentially turns the neural nets into an ODE and the optimization of the neural nets into an ODE initial value problem. The continuous hidden states can be efficiently evaluated by any ODE solver. Such a neural ODE model has the advantage of constant memory cost in terms of model depth and can trade-off precision and efficiency [49]. The adjoint method for differentiating the neural ODE is connected to our ADLES method.

While adjoint methods are efficient in high-dimensional neural models, discretization is often sufficient for low-dimensional ODEs. For example, [50] uses the Runge-Kutta method to discretize ODEs in time and minimize the estimation error via auto-differentiation. They obtain better performance in forecasting COVID-19 dynamics than NeuralODE.

Theoretical interpretation of deep neural models with dynamical systems Well-established theories in dynamical systems can help develop the theoretical framework and analytical tools for deep learning, such as [51, 52]. Particularly, there is a deep connection between dynamical systems and deep neural nets (DNNs). Dynamical systems bring differential geometric perspectives into the statistical learning regime, uniting two seemingly distant realms. With such union, various statistics, probability, geometry, topology, analysis, and algebra based tools can be employed to establish theoretical guarantees for deep neural nets in terms of deriving optimal solutions and

efficient computational/numerical methods [53, 54]. Since the theories in this realm are broad and deep, it is unrealistic and beyond our scope to enumerate them. Instead, we illustrate some facets of the key findings with two examples.

As one of the most successful neural net structures, deep residual networks [55] can be viewed as a flow map (see Section 6.1) on the phase/state space of model states. When made continuous in time, the deep residual nets become an ODE, similar to NeuralODE [49]. The training of deep residual nets is the discrete approximation of the continuous process, and the model output is the output of the dynamical system with input data as initial conditions [52]. Authors in [56] establish sufficient conditions for the universal approximation properties of such flow maps. Under mild regularity constraints or conditions, the flow maps can universally approximate any function with arbitrary precision defined in $\mathcal{L}^p$. The sufficient conditions are general without assuming specific layer structures.

Further, authors in [57] establish a flow representation of general DNNs, which formulate a DNN as the flow of an ODE. When discretized, the continuous flow of the dynamical system becomes the transport map, resulting in a general DNN architecture. In other words, *a DNN is the discretization of an ODE*. Consequently, from the perspective of optimal transport, a DNN optimization process is understood as finding the optimal transport map between two (source and target) probability distributions [57]. Such a flow representation framework can interpret various DNNs, such as residual networks, generative networks, and encoder-decoder networks. For example, a denoising auto-encoder is essentially the time reversal of a diffusion process—the backward heat equation. A graph neural net can be seen as a discrete diffusion process. Moreover, the flow representation provides a coordinate-free formulation of DNNs and helps reduce over-parameterization [57].

More generally, a DNN aims to *discover the topology of the data manifold* and *find the optimal transport map between distributions*. For the former, tools in the mathematical area of algebraic topology can help find the topological invariants under continuous transforms. For the latter, we can extend the flow and transport map between distributions to Wasserstein spaces. A Wasserstein space is a space of

probability distributions with Wasserstein geometry—that defines the Riemannian metric and covariant derivative of distributions. As a result, vector fields and gradient flows can describe how a distribution evolves in space and time, and the optimal transport map can be obtained via variational approaches [53, 54]. Namely, the optimal transport map is the gradient flow of some energy functional (e.g., entropy, divergence) and reduces to solving the Monge-Ampere differential equation [53, 54]. For instance, a denoising auto-encoder, which corresponds to a backward heat equation, is a Wasserstein gradient flow that increases the Shannon entropy functional [57].

Such dynamical system approaches give intuitive and theoretical interpretation to deep learning and foster new models and learning strategies, such as the Monge-Ampere flow model [58]. We will continue the discussion of machine learning approaches for solving PDEs in the next chapter.

6.5 Conclusions

This chapter studies the physical process of phonation and the process-specific modeling of the vocal fold and vocal tract. We present a dynamical system perspective for physical process modeling and phase space characterization of phonation. We propose a backward approach for modeling vocal fold dynamics and an efficient algorithm to solve the inverse problem of estimating model parameters from speech observations.

The oscillatory dynamics of vocal folds provide a tool to analyze different phonation phenomena, which characterize different types of voice disorders. We propose an ADLES method to promote accurate and efficient recovery of the parameters of an asymmetric vocal folds model directly from the speech signal. It allows us to correctly solve the oscillatory dynamics of the vocal folds for any specific speech signal. More importantly, the parameters estimated for the model directly allow us to predict voice pathology by simple analyses such as placement on the system's bifurcation map. They can also be potentially used to estimate the physical properties of the speaker's vocal folds. Moreover, the ADLES method significantly alleviates the difficulty of obtaining actual measurements of vocal fold displacements in clinical settings and can thus be a

valuable diagnostic aid for identifying different voice pathologies.

Lastly, we extend our process-specific models to deriving features for machine learning models and discuss the deep connection between deep neural models and dynamical systems. Dynamical system theories and methods provide valuable and insightful tools for advancing deep learning theories and applications, and these can, in turn, be used to improve process-specific modeling strategies.

References

[1] G. D. Birkhoff. *Dynamical systems*. Vol. 9. American Mathematical Soc., 1927.

[2] J. J. Jiang, Y. Zhang, and J. Stern. "Modeling of chaotic vibrations in symmetric vocal folds". In: *The Journal of the Acoustical Society of America* 110.4 (2001), pp. 2120–2128.

[3] J. J. Jiang and Y. Zhang. "Chaotic vibration induced by turbulent noise in a two-mass model of vocal folds". In: *The Journal of the Acoustical Society of America* 112.5 (2002), pp. 2127–2133.

[4] L. Cveticanin. "Review on Mathematical and Mechanical Models of the Vocal Cord". In: *Journal of Applied Mathematics* (2012).

[5] I. R. Titze. "The physics of small-amplitude oscillation of the vocal folds". In: *The Journal of the Acoustical Society of America* 83.4 (1988), pp. 1536–1552.

[6] R. Singh. *Profiling humans from their voice*. Springer, 2019.

[7] J. Flanagan and L. Landgraf. "Self-oscillating source for vocal-tract synthesizers". In: *IEEE Transactions on Audio and Electroacoustics* 16.1 (1968), pp. 57–64.

[8] K. Ishizaka and J. L. Flanagan. "Synthesis of voiced sounds from a two-mass model of the vocal cords". In: *Bell system technical journal* 51.6 (1972), pp. 1233–1268.

[9] Z. Zhang, J. Neubauer, and D. A. Berry. "The influence of subglottal acoustics on laboratory models of phonation". In: *The Journal of the Acoustical Society of America* 120.3 (2006), pp. 1558–1569.

[10] W. Zhao et al. “Computational aeroacoustics of phonation, Part I: Computational methods and sound generation mechanisms”. In: *The Journal of the Acoustical Society of America* 112.5 (2002), pp. 2134–2146.

[11] C. Zhang et al. “Computational aeroacoustics of phonation, Part II: Effects of flow parameters and ventricular folds”. In: *The Journal of the Acoustical Society of America* 112.5 (2002), pp. 2147–2154.

[12] J. C. Lucero. “Dynamics of the two-mass model of the vocal folds: Equilibria, bifurcations, and oscillation region”. In: *The Journal of the Acoustical Society of America* 94.6 (1993), pp. 3104–3111.

[13] J. C. Lucero and J. Schoentgen. “Modeling vocal fold asymmetries with coupled van der Pol oscillators”. In: *Proceedings of Meetings on Acoustics ICA2013*. Vol. 19. 1. ASA. 2013, p. 060165.

[14] F. Alipour, D. A. Berry, and I. R. Titze. “A finite-element model of vocal-fold vibration”. In: *The Journal of the Acoustical Society of America* 108.6 (2000), pp. 3003–3012.

[15] A. Yang et al. “Computation of physiological human vocal fold parameters by mathematical optimization of a biomechanical model”. In: *The Journal of the Acoustical Society of America* 130.2 (2011), pp. 948–964.

[16] B. A. Pickup and S. L. Thomson. “Influence of asymmetric stiffness on the structural and aerodynamic response of synthetic vocal fold models”. In: *Journal of biomechanics* 42.14 (2009), pp. 2219–2225.

[17] J. J. Jiang, Y. Zhang, and J. Stern. “Modeling of chaotic vibrations in symmetric vocal folds”. In: *Journal of the Acoustical Society of America* 10.4 (2001), pp. 2120–2128.

[18] B. H. Story and I. R. Titze. “Voice simulation with a body-cover model of the vocal folds”. In: *The Journal of the Acoustical Society of America* 97.2 (1995), pp. 1249–1260.

[19] R. W. Chan and I. R. Titze. "Dependence of phonation threshold pressure on vocal tract acoustics and vocal fold tissue mechanics". In: *The Journal of the Acoustical Society of America* 119.4 (2006), pp. 2351–2362.

[20] J. C. Lucero et al. "Self-entrainment of the right and left vocal fold oscillators". In: *The Journal of the Acoustical Society of America* 137.4 (2015), pp. 2036–2046.

[21] S. Maeda. "Compensatory articulation during speech: Evidence from the analysis and synthesis of vocal-tract shapes using an articulatory model". In: *Speech production and speech modelling*. Springer, 1990, pp. 131–149.

[22] P. Birkholz and B. J. Kröger. "Simulation of vocal tract growth for articulatory speech synthesis". In: *Proceedings of the 16th international congress of phonetic sciences*. 2007, pp. 377–380.

[23] J. Dang and K. Honda. "Construction and control of a physiological articulatory model". In: *The Journal of the Acoustical Society of America* 115.2 (2004), pp. 853–870.

[24] M. R. Portnoff. "A quasi-one-dimensional digital simulation for the time-varying vocal tract." PhD thesis. Massachusetts Institute of Technology, 1973.

[25] D. R. Allen and W. J. Strong. "A model for the synthesis of natural sounding vowels". In: *The Journal of the Acoustical Society of America* 78.1 (1985), pp. 58–69.

[26] K. Motoki et al. "Computation of 3-D vocal tract acoustics based on mode-matching technique". In: *Sixth International Conference on Spoken Language Processing*. 2000.

[27] I. R. Titze and D. W. Martin. *Principles of voice production*. 1998.

[28] C. Bhat and S. K. Kopparapu. "FEMH Voice Data Challenge: Voice disorder Detection and Classification using Acoustic Descriptors". In: *2018 IEEE International Conference on Big Data (Big Data)*. IEEE. 2018, pp. 5233–5237.

[29] I. Steinecke and H. Herzel. "Bifurcations in an asymmetric vocal-fold model". In: *The Journal of the Acoustical Society of America* 97.3 (1995), pp. 1874–1884.

[30] B. D. Erath and M. W. Plesniak. “An investigation of jet trajectory in flow through scaled vocal fold models with asymmetric glottal passages”. In: *Experiments in fluids* 41.5 (2006), pp. 735–748.

[31] P. Mergell, H. Herzel, and I. R. Titze. “Irregular vocal-fold vibration—High-speed observation and modeling”. In: *The Journal of the Acoustical Society of America* 108.6 (2000), pp. 2996–3002.

[32] C. Tao et al. “Asymmetric airflow and vibration induced by the Coanda effect in a symmetric model of the vocal folds”. In: *The Journal of the Acoustical Society of America* 122.4 (2007), pp. 2270–2278.

[33] V. Isakov. *Inverse problems for partial differential equations*. Vol. 127. Springer, 2006.

[34] C. Tao et al. “Estimating model parameters by chaos synchronization”. In: *Physical Review E* 69.3 (2004), p. 036204.

[35] Y. Zhang, C. Tao, and J. J. Jiang. “Parameter estimation of an asymmetric vocal-fold system from glottal area time series using chaos synchronization”. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 16.2 (2006), p. 023118.

[36] S. J. Rupitsch et al. “Simulation based estimation of dynamic mechanical properties for viscoelastic materials used for vocal fold models”. In: *Journal of Sound and Vibration* 330.18-19 (2011), pp. 4447–4459.

[37] C. Tao, Y. Zhang, and J. J. Jiang. “Extracting physiologically relevant parameters of vocal folds from high-speed video image series”. In: *IEEE Transactions on Biomedical Engineering* 54.5 (2007), pp. 794–801.

[38] P. Birkholz, D. Jackèl, and B. J. Kroger. “Construction and control of a three-dimensional vocal tract model”. In: *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*. Vol. 1. IEEE. 2006, pp. I–I.

[39] J. Mullen, D. M. Howard, and D. T. Murphy. “Real-time dynamic articulations in the 2-D waveguide mesh vocal tract model”. In: *IEEE Transactions on Audio, Speech, and Language Processing* 15.2 (2007), pp. 577–585.

[40] W. Zhao and R. Singh. "Speech-based parameter estimation of an asymmetric vocal fold oscillation model and its application in discriminating vocal fold pathologies". In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2020, pp. 7344–7348.

[41] M. M. Sondhi. "Model for wave propagation in a lossy vocal tract". In: *The Journal of the Acoustical Society of America* 55.5 (1974), pp. 1070–1075.

[42] P. Alku. "Glottal inverse filtering analysis of human voice production—A review of estimation and parameterization methods of the glottal excitation and their applications". In: *Sadhana* 36.5 (2011), pp. 623–650.

[43] M. Al Ismail, S. Deshmukh, and R. Singh. "Detection of COVID-19 through the analysis of vocal fold oscillations". In: *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2021, pp. 1035–1039.

[44] S. Deshmukh, M. Al Ismail, and R. Singh. "Interpreting glottal flow dynamics for detecting covid-19 from voice". In: *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2021, pp. 1055–1059.

[45] J. Zhang. "Vocal Fold Dynamics for Automatic Detection of Amyotrophic Lateral Sclerosis from Voice". BA thesis. Carnegie Mellon University, USA: Computational Biology Department, May 2022.

[46] R. Wang and R. Yu. "Physics-guided deep learning for dynamical systems: A survey". In: *arXiv preprint arXiv:2107.01272* (2021).

[47] G. Liao and L. Zhang. "Solving flows of dynamical systems by deep neural networks and a novel deep learning algorithm". In: *Mathematics and Computers in Simulation* (2022).

[48] P. Rajendra and V. Brahmajirao. "Modeling of dynamical systems through deep learning". In: *Biophysical Reviews* 12.6 (2020), pp. 1311–1320.

[49] R. T. Chen et al. “Neural ordinary differential equations”. In: *Advances in neural information processing systems* 31 (2018).

[50] R. Wang et al. “Bridging physics-based and data-driven modeling for learning dynamical systems”. In: *Learning for Dynamics and Control*. PMLR. 2021, pp. 385–398.

[51] S. Sonoda and N. Murata. “Double continuum limit of deep neural networks”. In: *ICML Workshop Principled Approaches to Deep Learning*. Vol. 1740. 2017.

[52] M. Thorpe and Y. van Gennip. “Deep limits of residual neural networks”. In: *arXiv preprint arXiv:1810.11741* (2018).

[53] N. Lei et al. “A geometric view of optimal transportation and generative model”. In: *Computer Aided Geometric Design* 68 (2019), pp. 1–21.

[54] X. Gu, N. Lei, and S.-T. Yau. “Optimal Transport for Generative Models”. In: *Handbook of Mathematical Models and Algorithms in Computer Vision and Imaging: Mathematical Imaging and Vision*. Springer, 2021, pp. 1–48.

[55] K. He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[56] Q. Li, T. Lin, and Z. Shen. “Deep learning via dynamical systems: An approximation perspective”. In: *Journal of the European Mathematical Society* (2022).

[57] S. Sonoda and N. Murata. “Transport analysis of infinitely deep neural network”. In: *The Journal of Machine Learning Research* 20.1 (2019), pp. 31–82.

[58] L. Zhang, L. Wang, et al. “Monge-amp\ere flow for generative modeling”. In: *arXiv preprint arXiv:1809.10188* (2018).

Chapter 7

Process-Specific Approaches for Vocal Tract Modeling

In the previous chapter, we described the process-specific modeling for phonation—i.e., specifically for the self-sustained motion of the vocal folds. We proposed a backward approach to solve the coupled ODE systems and the ADLES method for efficiently estimating the model parameters and characterizing disordered voices. This chapter extends the process-specific modeling to the combined vocal fold and vocal tract system and presents a forward-backward paradigm for solving the corresponding coupled ODE-PDE systems. In this chapter, we also extend the ADLES method to solve the inverse problem of estimating vocal fold-tract model parameters from observations and present an efficient algorithm with numerical solutions.

7.1 Modeling Wave Propagation in the Vocal Tract

The vocal tract can be viewed as a compact, orientable, differentiable manifold M embedded in $\mathbb{R}^3$. Its boundary ∂M includes the wall of the vocal tract. Consider the tangent bundle TM. Denote the set of all vector fields on TM as $\Gamma(TM)$, which is a $C^\infty(M)$-module [1]. A vector field is a smooth section on $TM, \Gamma(TM) \ni \boldsymbol{X} : M \to TM$. It associates each point $\boldsymbol{p} \in M$ with a tangent vector $\bar{\boldsymbol{v}}(\boldsymbol{p}) \coloneqq \boldsymbol{X}|_{\boldsymbol{p}} : C^\infty(M) \xrightarrow{\sim} \mathbb{R}$ [1]. Let $\gamma(t) : \mathbb{R} \supseteq I \to M$ be a maximal integral curve [1] through $\boldsymbol{p}$ at t_0 which is a

Table 7.1: Symbol List

Symbol	Description
ξ_0	displacement of left/right vocal fold at rest position from center line
$\xi_{l,r}$	displacement of left, right vocal fold from rest position
d	length of vocal fold
α	subglottal pressure coupling coefficient
β	combined mass, damping, spring coefficient
Δ	asymmetry coefficient
M	differentiable manifold
∂M	boundary of M
TM	tangent bundle
$C^\infty(M)$	ring of smooth functions over M
$\Gamma(TM)$	the set of vector fields on M, a $C^\infty(M)$-module
$\boldsymbol{X}$	vector field, a smooth section of TM
$\gamma(t)$	integral curve on M
Φ	flow of $\boldsymbol{X}$
$\boldsymbol{p}$	a point on M
$\bar{\boldsymbol{v}}(\boldsymbol{p})$	$= \boldsymbol{X}\|_{\boldsymbol{p}}$, tangent vector at $\boldsymbol{p}$
$\boldsymbol{v}(\boldsymbol{p}, t)$	air particle velocity at position $\boldsymbol{p}$ and time t
$\hat{p}(\boldsymbol{p}, t)$	acoustic pressure at position $\boldsymbol{p}$ and time t
$p(x, t)$	average acoustic pressure at position x and time t
$u(x, t)$	volume velocity at position x and time t
$z(x, t)$	time reversed volume velocity of $u(x, t)$
$f(x, t)$	vocal tract characteristic profile
Σ	inner surface of vocal tract
$R(\cdot)$	shape function of Σ
$A(\cdot)$	area function of Σ
c	speed of sound
$\tilde{c}$	air particle velocity at the midpoint of the vocal fold
ρ	ambient air density
Ω	spatial domain of acoustic wave
Γ	boundary of Ω
L	length of vocal tract
T	time domain of acoustic wave
t_m	maximum of T
p_m	measured acoustic pressure at lip
u_m	measured volume velocity at lip
$\mathcal{L}^n$	space of n-th Lebesgue integrable functions
$\mathcal{W}^{k,n}$	Sobolev space of order k
$\mathcal{H}$	nonlinear operator from input u_0 to output u_L
$\mathcal{F}$	nonlinear operator from f to u
$\mathcal{F}^*$	adjoint operator of $\mathcal{F}$
$\mathcal{L}$	Lagrangian
λ, η, μ, ν	Lagrangian multipliers
D_t	finite difference operator in time
R^n	PDE residual at time step n
v, w	test functions
$a(\cdot, \cdot)$, $L(\cdot)$	variational forms
P_k	triangular finite element of order k

solution to

$$\gamma'(t) = \boldsymbol{X}(\gamma(t))$$
$$\gamma(t_0) = \boldsymbol{p}$$

The curve $\gamma(t)$ is a one-parameter group. When acting on the Lie group M, it gives the flow $\Phi : \mathbb{R} \times M \to M$. $\Phi_t(\boldsymbol{p}) = \gamma(t)$. The particle velocity at $\boldsymbol{p}$ is given by $\boldsymbol{v}(\boldsymbol{p}, t) \coloneqq \gamma'(t) = \bar{\boldsymbol{v}}(\boldsymbol{p}) \circ \gamma(t)$. The planar motion of the pressure wave in the vocal tract is governed by the equations [2]

$$\frac{1}{\rho c^2}\frac{\partial \hat{p}}{\partial t} + \mathrm{div}\boldsymbol{v} = 0 \tag{7.1}$$
$$\rho\frac{\partial \boldsymbol{v}}{\partial t} + \mathrm{grad}\hat{p} = 0 \tag{7.2}$$

where $\hat{p}(\boldsymbol{p}, t)$ is the acoustic pressure, div is the divergence operator, grad is the gradient operator, ρ is the ambient air density, and c is the speed of sound. Equation (7.1) describes the conservation of mass, and (7.2) describes the conservation of momentum [2]. For notational convenience, we adapt cylindrical coordinates $\boldsymbol{p} = (r, \theta, x)$, where the x direction aligns with the central axis of vocal tract. We denote the inner surface of the vocal tract as Σ, and the shape function of the inner surface as $r = R(\theta, x)$. Then the cross-sectional area of the vocal tract is

$$A(x) = \int_0^{2\pi} d\theta \int_0^{R(\theta,x)} r dr \tag{7.3}$$

the average acoustic pressure is

$$p(x, t) = \frac{1}{A(x)} \int_0^{2\pi} d\theta \int_0^{R(\theta,x)} \hat{p} r dr \tag{7.4}$$

and the volume velocity is

$$u(x, t) = \int_0^{2\pi} d\theta \int_0^{R(\theta,x)} v_x r dr \tag{7.5}$$

where v_x is the x component of $\boldsymbol{v}$. Integrating (7.1) over the volume of vocal tract bounded by cross sections at x_0 and x gives

$$0 = \int_M \frac{1}{\rho c^2}\frac{\partial \hat{p}}{\partial t} + \text{div}\boldsymbol{v} \tag{7.6}$$

$$= \int_{x_0}^{x}\left[\int_0^{2\pi} d\theta \int_0^R \frac{1}{\rho c^2}\frac{\partial \hat{p}}{\partial t} r dr\right] dx' + \int_M \text{div}\boldsymbol{v} \tag{7.7}$$

$$= \frac{1}{\rho c^2}\int_{x_0}^{x} A(x')\frac{\partial p(x',t)}{\partial t}dx' + \iint_\Sigma n_{\boldsymbol{v}} d\sigma + u(x,t) - u(x_0,t) \tag{7.8}$$

where from (7.7) to (7.8) we substitute into (7.4), (7.5) and apply Stokes' theorem [2, 3]; $n_{\boldsymbol{v}}$ is the component of $\boldsymbol{v}$ normal and outward to the inner surface Σ. The element of area $d\sigma$ is given by [2, 3]

$$d\sigma = S(\theta, x)d\theta dx \tag{7.9}$$

where $Sd\theta dx$ is a top 2-form on Σ [1]. Substituting (7.9) into (7.8) and differentiating w.r.t. x yields

$$\frac{A(x)}{\rho c^2}\frac{\partial p}{\partial t} + \frac{\partial u}{\partial x} + \int_0^{2\pi} n_{\boldsymbol{v}}(\theta, x, t)S(\theta, x)d\theta = 0 \tag{7.10}$$

Following similar steps, integrating the x component of (7.2) over the cross section at x yields

$$\rho\frac{\partial u}{\partial t} + A(x)\frac{\partial p}{\partial x} + \int_0^{2\pi}\left(p(x,t) - p_w(\theta, x, t)\right)\frac{\partial}{\partial x}\left(\frac{1}{2}R^2\right)d\theta = 0 \tag{7.11}$$

where p_w is the pressure acting on the wall of the vocal tract.

7.1.1 Integrated Vocal Tract Model

To simplify our problem, we combine the wave equations (7.10) and (7.11) into a single vocal tract model. Differentiating (7.10) w.r.t x and (7.11) w.r.t. t, and cancelling out

the pressure term gives

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2} + \frac{1}{\rho}\frac{\partial A}{\partial x}\frac{\partial p}{\partial t} - \frac{1}{\rho}\partial_t \int_0^{2\pi} \left(p(x,t) - p_w(\theta,x,t)\right) \frac{\partial}{\partial x}\left(\frac{1}{2}R^2\right) d\theta$$
$$+ c^2 \partial_x \int_0^{2\pi} n_{\boldsymbol{v}}(\theta,x,t) S(\theta,x) d\theta \tag{7.12}$$
$$= c^2 \frac{\partial^2 u}{\partial x^2} + f(x,t) \tag{7.13}$$

where the vocal tract profile is absorbed into a single term $f(x,t)$. It represents the characteristics of the vocal tract, i.e., the effect of the nonuniform yielding wall on the acoustic flow dynamics, which needs to be estimated by our algorithm.

7.2 Parameter Estimation for Vocal Fold-Tract Model

7.2.1 Problem Formulation

We now formulate the problem of estimating the parameters of the combined vocal fold-tract model from speech measurements. Let $\Omega \times T$ be the domain of volume velocity u, where Ω is the spatial domain, and T is the time domain. In the one-dimensional case, $\Omega = [0, L]$ where L is the length of vocal tract, and $T = [0, t_m]$, where t_m is the maximum of T. Given a measured acoustic pressure $p_m(t)$ at the lip, the corresponding volume velocity is given by [4]

$$u_m(t) = \frac{A(L)}{\rho c} p_m(t) \tag{7.14}$$

where $A(L)$ is the opening area at the lip, c is the speed of sound, and ρ is the ambient air density. Denote $u_0(t) \coloneqq u(0,t)$, $u_L(t) \coloneqq u(L,t)$. The glottal flow $u_0(t)$ can be derived from the vocal folds displacement model (6.3) by

$$u_0(t) = \tilde{c} d (2\xi_0 + \xi_l(t) + \xi_r(t)) \tag{7.15}$$

where ξ_0 is the half glottal width at rest, d is the length of the vocal fold, and $\tilde{c}$ is the air particle velocity at the midpoint of the vocal fold (see Figure 6-4). Let $\mathcal{H}$ be the

nonlinear operator representing acoustic wave propagation from the glottis to the lip. We have the forward propagation process as

$$\begin{aligned} \mathcal{H} : \mathcal{L}^2(\Omega \times T) \times \mathcal{L}^2(\Gamma \times T) &\to \mathcal{L}^2(\Gamma \times T) \\ (f, u_0) &\mapsto u_L \end{aligned} \tag{7.16}$$

where f is the vocal tract profile in (7.13), and $\Gamma = \partial\Omega$ is the boundary. We can split Γ into two parts: $\Gamma = \Gamma_0 \bigcup \Gamma_1$, and $\Gamma_0 \bigcap \Gamma_L = \emptyset$ corresponding to $x = 0$ and $x = L$. However, we disregard the difference to simplify our derivation. Note that in the one-dimensional case, $u(t)$ and $u_L(t)$ are only functions of t. However, more generally, they are functions of both x on the boundary Γ and t. We define two nonlinear operators as

$$\begin{aligned} \mathcal{H}_f : \mathcal{L}^2(\Gamma \times T) &\to \mathcal{L}^2(\Gamma \times T) \\ u_0 &\mapsto u_L \end{aligned} \tag{7.17}$$

$$\begin{aligned} \mathcal{F} \coloneqq \mathcal{H}_{u_0} : \mathcal{L}^2(\Omega \times T) &\to \mathcal{L}^2(\Gamma \times T) \\ f &\mapsto u_L \end{aligned} \tag{7.18}$$

Note that both $\mathcal{H}_f$ and $\mathcal{F}$ are bounded. Our objective is to minimize the difference between the measured volume velocity u_m and the predicted volume velocity u_L near

the lip, subject to the constraints

$$\min \int_0^T \left(\mathcal{H}_f(u_0(t)) - u_m(t)\right)^2 dt \tag{7.19}$$

$$\Leftrightarrow \min \int_0^T \left(\mathcal{H}_f(\tilde{c}d(2\xi_0 + \xi_l(t) + \xi_r(t))) - \frac{A(L)}{\rho c} p_m(t)\right)^2 dt \tag{7.20}$$

$$\text{subject to} \quad \ddot{\xi}_r + \beta(1+\xi_r^2)\dot{\xi}_r + \xi_r - \frac{\Delta}{2}\xi_r = \alpha(\dot{\xi}_r + \dot{\xi}_l) \tag{7.21}$$

$$\ddot{\xi}_l + \beta(1+\xi_l^2)\dot{\xi}_l + \xi_l + \frac{\Delta}{2}\xi_l = \alpha(\dot{\xi}_r + \dot{\xi}_l) \tag{7.22}$$

$$\text{(I.C.1)} \quad \xi_r(0) = C_r \tag{7.23}$$

$$\text{(I.C.2)} \quad \xi_l(0) = C_l \tag{7.24}$$

$$\text{(I.C.3)} \quad \dot{\xi}_r(0) = 0 \tag{7.25}$$

$$\text{(I.C.4)} \quad \dot{\xi}_l(0) = 0 \tag{7.26}$$

$$\tag{7.27}$$

where (7.21) and (7.22) represent the asymmetric vocal folds displacement model (6.3), I.C. stands for initial condition, and Cs are constants. Next, we derive an efficient strategy to estimate the parameters α, β, and Δ such that (7.20) is minimized.

7.2.2 Solving Vocal Tract Model via Forward-Backward Method

In order to solve the parameter estimation problem (7.27), first, we need to estimate the vocal tract profile f in $\mathcal{H}_f$ and (7.13). Specifically, we need to solve

$$\frac{\partial^2 u(x,t)}{\partial t^2} = c^2 \frac{\partial^2 u(x,t)}{\partial x^2} + f(x,t) \tag{7.28}$$

subject to

$$\text{(B.C.1)} \quad u(0,t) = u_g(t) \tag{7.29}$$

$$\text{(B.C.2)} \quad u(L,t) = u_m(t) \tag{7.30}$$

$$\text{(B.C.3)} \quad \frac{\partial u}{\partial n_\Gamma} = 0 \tag{7.31}$$

$$\text{(I.C.1)} \quad u(x,0) = 0 \tag{7.32}$$

$$\text{(I.C.2)} \quad \frac{\partial u(x,0)}{\partial t} = 0 \tag{7.33}$$

$$\tag{7.34}$$

where B.C. stands for boundary condition, u_g and u_m are volume velocity at the glottis and lip, respectively, and n_Γ is the outward unit normal to the boundary Γ. We now derive the solution to (7.34). In order to estimate $f \in \mathcal{L}^2(\Omega \times T)$, we take an iterative approach, i.e.

$$f^{k+1} = f^k + \tau \delta f^k \tag{7.35}$$

where $\delta f^k \in \mathcal{L}^2(\Omega \times T)$ is a small variation, and τ is a step size. Taking the Taylor expansion of $\mathcal{F}$ (7.18) at f^k gives

$$\mathcal{F}(f^k + \delta f^k) = \mathcal{F}(f^k) + \mathcal{F}'(f^k)\delta f^k + O\left((\delta f^k)^2\right) \tag{7.36}$$

where $\mathcal{F}'$ is the Fréchet derivative [5]. Omitting higher order terms, we obtain

$$\mathcal{F}'(f^k)\delta f^k = \mathcal{F}(f^k + \delta f^k) - \mathcal{F}(f^k) \tag{7.37}$$

where $\mathcal{F}'(f)$ is a nonlinear operator

$$\begin{aligned} \mathcal{F}'(f) : \mathcal{L}^2(\Omega \times T) &\to \mathcal{L}^2(\Gamma \times T) \\ \delta f &\mapsto \delta u_L \end{aligned} \tag{7.38}$$

Correspondingly, the adjoint operator [5, 6, 7] is

$$\begin{aligned} \mathcal{F}'(f)^* : \mathcal{L}^2(\Gamma \times T) &\to \mathcal{L}^2(\Omega \times T) \\ \delta u_L &\mapsto \delta f \end{aligned} \tag{7.39}$$

We would like $\mathcal{F}(f^k + \delta f^k) = u_L^k + \delta u_L^k \xrightarrow{k\to\infty} u_m$. This is equivalent to solving

$$\begin{aligned} &\min \ \|\delta f^k\|_2^2 \\ \text{subject to} \quad &\mathcal{F}'(f^k)\delta f^k = u_m - \mathcal{F}(f^k) \end{aligned} \tag{7.40}$$

It is simple to obtain the solution to (7.40)

$$\delta f^k = -\mathcal{F}'(f^k)^* \left[\mathcal{F}'(f^k)\mathcal{F}'(f^k)^*\right]^{-1} \left(\mathcal{F}(f^k) - u_m\right) \tag{7.41}$$

where $\mathcal{F}'(f^k)^*$ is the adjoint operator. It is difficult to compute $\mathcal{F}'(f^k)\mathcal{F}'(f^k)^*$. By positive-definiteness, we approximate it by $\gamma \boldsymbol{I}$ where $\boldsymbol{I}$ is the identity matrix. We denote the estimation residual as

$$r^k \coloneqq u_m - \mathcal{F}(f^k) \tag{7.42}$$

We now have

$$\delta f^k = \tfrac{1}{\gamma}\mathcal{F}'(f^k)^* r^k \tag{7.43}$$

Now consider the wave equation (7.28). Let $u + \delta u$ be a solution with variation $f + \delta f$. Substitution into (7.28) yields

$$\frac{\partial^2(u+\delta u)}{\partial t^2} = c^2 \frac{\partial^2(u+\delta u)}{\partial x^2} + f + \delta f \tag{7.44}$$

Subtracting (7.28) yields

$$\frac{\partial^2 \delta u}{\partial t^2} = c^2 \frac{\partial^2 \delta u}{\partial x^2} + \delta f \tag{7.45}$$

subject to

$$\text{(B.C.1)} \quad \frac{\partial \delta u}{\partial n_\Gamma} = 0 \tag{7.46}$$

$$\text{(I.C.1)} \quad \delta u(x,0) = 0 \tag{7.47}$$

$$\text{(I.C.2)} \quad \frac{\partial \delta u(x,0)}{\partial t} = 0 \tag{7.48}$$

$$\tag{7.49}$$

Next, we use a time-reversal technique [2] and backpropagate the difference (7.42) into the vocal tract, which gives

$$\frac{\partial^2 z}{\partial t^2} = c^2 \frac{\partial^2 z}{\partial x^2} + f(x,t) \tag{7.50}$$

subject to

$$\text{(B.C.1)} \quad \frac{\partial z}{\partial n_\Gamma} = r \tag{7.51}$$

$$\text{(I.C.1)} \quad z(x,t_m) = 0 \tag{7.52}$$

$$\text{(I.C.2)} \quad \frac{\partial z(x,t_m)}{\partial t} = 0 \tag{7.53}$$

$$\tag{7.54}$$

where z is the time reversal of u. It follows [8] that

$$\langle \delta f, z \rangle_{\Omega \times T} = \int_0^{t_m} \int_\Omega \delta f z dx dt \tag{7.55}$$

$$= \int_0^{t_m} \int_\Omega \left(\frac{\partial^2 \delta u}{\partial t^2} - c^2 \frac{\partial^2 \delta u}{\partial x^2} \right) z dx dt \tag{7.56}$$

$$= \int_0^{t_m} \int_\Omega \left(\frac{\partial^2 \delta u}{\partial t^2} - c^2 \frac{\partial^2 \delta u}{\partial x^2} \right) z dx dt - \int_0^{t_m} \int_\Omega \left(\frac{\partial^2 z}{\partial t^2} - c^2 \frac{\partial^2 z}{\partial x^2} - f \right) \delta u dx dt \tag{7.57}$$

$$= \int_0^{t_m} \int_\Omega \left(\frac{\partial^2 \delta u}{\partial t^2} z - \frac{\partial^2 z}{\partial t^2} \delta u \right) dx dt - c^2 \int_0^{t_m} \int_\Omega \left(\frac{\partial^2 \delta u}{\partial x^2} z - \frac{\partial^2 z}{\partial x^2} \delta u \right) dx dt + \int_0^{t_m} \int_\Omega f \delta u dx dt \tag{7.58}$$

$$= \int_\Omega \left(\frac{\partial \delta u}{\partial t} z - \frac{\partial z}{\partial t} \delta u \right) \Bigg|_0^{t_m} dx dt - c^2 \int_0^{t_m} \int_\Omega \left(\frac{\partial^2 \delta u}{\partial x^2} z - \frac{\partial^2 z}{\partial x^2} \delta u \right) dx dt + \int_0^{t_m} \int_\Omega f \delta u dx dt \tag{7.59}$$

$$= -c^2 \int_0^{t_m} \int_\Omega \left(\frac{\partial^2 \delta u}{\partial x^2} z - \frac{\partial^2 z}{\partial x^2} \delta u \right) dx dt + \int_0^{t_m} \int_\Omega f \delta u dx dt \tag{7.60}$$

$$= -c^2 \int_0^{t_m} \int_\Omega \left(z d \frac{\partial \delta u}{\partial x} - \delta u d \frac{\partial z}{\partial x} \right) dt + \int_0^{t_m} \int_\Omega f \delta u dx dt \tag{7.61}$$

$$= -c^2 \int_0^{t_m} \left(\int_\Gamma z \frac{\partial \delta u}{\partial n_\Gamma} ds - \int_\Omega \frac{\partial \delta u}{\partial x} \frac{\partial z}{\partial x} dx - \int_\Gamma \delta u \frac{\partial z}{\partial n_\Gamma} ds + \int_\Omega \frac{\partial \delta u}{\partial x} \frac{\partial z}{\partial x} dx \right) dt + \int_0^{t_m} \int_\Omega f \delta u dx dt \tag{7.62}$$

$$= c^2 \int_0^{t_m} \int_\Gamma \delta u \frac{\partial z}{\partial n_\Gamma} ds dt + \int_0^{t_m} \int_\Omega f \delta u dx dt \tag{7.63}$$

$$= c^2 \int_0^{t_m} \int_\Gamma \delta u r ds dt + \int_0^{t_m} \int_\Omega f \delta u dx dt \tag{7.64}$$

$$= c^2 \int_0^{t_m} \int_\Gamma \mathcal{F}'(f) \delta f r ds dt + \int_0^{t_m} \int_\Omega f \delta u dx dt \tag{7.65}$$

$$= c^2 \int_0^{t_m} \int_\Omega \delta f \mathcal{F}'(f)^* r dx dt + \int_0^{t_m} \int_\Omega f \delta u dx dt \tag{7.66}$$

$$= c^2 \int_0^{t_m} \int_\Omega \delta f \mathcal{F}'(f)^* r dx dt - \int_0^{t_m} \int_\Omega \delta f u dx dt \tag{7.67}$$

$$= c^2 \int_0^{t_m} \int_\Omega \delta f \left(\mathcal{F}'(f)^* r - u \right) dx dt \tag{7.68}$$

wherein from (7.55) to (7.57) we substitute into (7.45) and (7.50); from (7.57) to (7.60) we apply initial conditions (7.47), (7.48), (7.52) and (7.53); from (7.60) to

(7.62) we integrate by parts; from (7.62) to (7.63) we apply boundary condition (7.46); from (7.63) to (7.64) we use boundary condition (7.51); from (7.64) to (7.65) we use definition (7.38); from (7.65) to (7.66) we use definition (7.39) and the duality property

$$\langle \mathcal{F}'(f)\delta f, r\rangle_{\Gamma\times T} = \langle \delta f, \mathcal{F}'(f)^* r\rangle_{\Omega\times T}$$

from (7.66) to (7.67) we assume the second-order variation is small, i.e.

$$\langle f+\delta f, u+\delta u\rangle = \langle f, u\rangle + \langle f, \delta u\rangle + \langle \delta f, u\rangle + \langle \delta f, \delta u\rangle \approx \langle f, u\rangle$$

(or $\delta(fu) = \delta(f)u + f\delta(u) \approx 0$.) By the arbitrariness of δf, it follows that

$$z = c^2(\mathcal{F}'(f)^* r - u)$$

and hence

$$\mathcal{F}'(f)^* r = \frac{z}{c^2} + u \tag{7.69}$$

Substitution into (7.43) and (7.35) yields

$$f^{k+1} = f^k + \frac{\tau}{\gamma}\left(\frac{z^k}{c^2} + u^k\right) \tag{7.70}$$

Hence, we obtain an iterative forward-backward approach for solving the vocal tract profile f.

7.2.3 Estimating Model Parameters via Adjoint Least Squares Method

Now, we derive the solution to the parameter estimation problem in (7.27), using the adjoint least squares method proposed in Section 6.3.3. Denote the estimation error as

$$f(\xi_l, \xi_r; \vartheta) = \left(\mathcal{H}_f(\tilde{c}d(2\xi_0 + \xi_l(t) + \xi_r(t))) - \frac{A(L)}{\rho c} p_m(t)\right)^2$$

and

$$F(\xi_l, \xi_r; \vartheta) = \int_0^{t_m} f(\xi_l, \xi_r; \vartheta) dt$$

where $\vartheta = [\alpha, \beta, \Delta]$ are the parameters in the vocal folds model (6.3). We would like to obtain the update rules for the model parameters α, β, and Δ, i.e.

$$\alpha^{k+1} = \alpha^k - \tau^\alpha F_{\alpha^k} \tag{7.71}$$

$$\beta^{k+1} = \beta^k - \tau^\beta F_{\beta^k} \tag{7.72}$$

$$\Delta^{k+1} = \Delta^k - \tau^\Delta F_{\Delta^k} \tag{7.73}$$

$$\tag{7.74}$$

where the partial derivatives $F_\cdot := \partial_\cdot F \equiv \frac{\partial F}{\partial \cdot}$ and $\tau^\cdot$ is the step size. We define the Lagrangian

$$\begin{aligned}\mathcal{L}(\vartheta) = \int_0^{t_m} \Bigg[& f + \lambda \left(\ddot{\xi}_r + \beta(1+\xi_r^2)\dot{\xi}_r + \xi_r - \frac{\Delta}{2}\xi_r - \alpha(\dot{\xi}_r + \dot{\xi}_l) \right) \\ & + \eta \left(\ddot{\xi}_l + \beta(1+\xi_l^2)\dot{\xi}_l + \xi_l + \frac{\Delta}{2}\xi_l - \alpha(\dot{\xi}_r + \dot{\xi}_l) \right) \Bigg] dt \\ & + \mu_l(\xi_l(0) - C_l) + \mu_r(\xi_r(0) - C_r) + \nu_l \dot{\xi}_l(0) + \nu_r \dot{\xi}_r(0)\end{aligned} \tag{7.75}$$

where λ, η, μ and ν are multipliers. Taking the derivative of the Lagrangian w.r.t. the model parameter α yields

$$\begin{aligned}\mathcal{L}_\alpha = \int_0^{t_m} \Bigg[& 2\tilde{c}d\mathcal{H}_f'\Big|_{u_0} (\partial_\alpha \xi_l + \partial_\alpha \xi_r) \\ & + \lambda \left(\partial_\alpha \ddot{\xi}_r + 2\beta \dot{\xi}_r \xi_r \partial_\alpha \xi_r + \beta(1+\xi_r^2)\partial_\alpha \dot{\xi}_r + \partial_\alpha \xi_r - \frac{\Delta}{2}\partial_\alpha \xi_r - \alpha(\partial_\alpha \dot{\xi}_r + \partial_\alpha \dot{\xi}_l) - (\dot{\xi}_r + \dot{\xi}_r) \right) \\ & + \eta \left(\partial_\alpha \ddot{\xi}_l + 2\beta \dot{\xi}_l \xi_l \partial_\alpha \xi_l + \beta(1+\xi_l^2)\partial_\alpha \dot{\xi}_l + \partial_\alpha \xi_l + \frac{\Delta}{2}\partial_\alpha \xi_l - \alpha(\partial_\alpha \dot{\xi}_r + \partial_\alpha \dot{\xi}_l) - (\dot{\xi}_r + \dot{\xi}_r) \right) \Bigg] dt \\ & + \mu_l \partial_\alpha \xi_l(0) + \mu_r \partial_\alpha \xi_r(0) + \nu_l \partial_\alpha \dot{\xi}_l(0) + \nu_r \partial_\alpha \dot{\xi}_r(0)\end{aligned} \tag{7.76}$$

Integrating the term $\lambda \partial_\alpha \ddot{\xi}_r$ by parts twice gives

$$\int_0^{t_m} \lambda \partial_\alpha \ddot{\xi}_r dt = \int_0^{t_m} \partial_\alpha \xi_r \ddot{\lambda} dt - \partial_\alpha \xi_r \dot{\lambda}\Big|_0^{t_m} + \partial_\alpha \dot{\xi}_r \lambda\Big|_0^{t_m} \tag{7.77}$$

Define the estimation residual $R \coloneqq \mathcal{H}_f(u_0) - \frac{A(L)}{\rho c} p_m(t)$. Applying the same to $\eta \partial_\alpha \ddot{\xi}_l$, substitution into (7.76) and subsequent simplification yields

$$\begin{aligned}
\mathcal{L}_\alpha = \int_0^{t_m} \Bigg[& \left(\ddot{\lambda} + \left(2\beta \xi_r \dot{\xi}_r + 1 - \frac{\Delta}{2} \right) \lambda + 2\tilde{c} dR \mathcal{H}_f' \Big|_{u_0} \right) \partial_\alpha \xi_r \\
& + \left(\ddot{\eta} + \left(2\beta \xi_l \dot{\xi}_l + 1 + \frac{\Delta}{2} \right) \lambda + 2\tilde{c} dR \mathcal{H}_f' \Big|_{u_0} \right) \partial_\alpha \xi_l \\
& + \Big(\beta(1+\xi_r^2)\lambda - \alpha(\lambda+\eta) \Big) \partial_\alpha \dot{\xi}_r + \Big((\beta(1+\xi_l^2)\eta - \alpha(\lambda+\eta) \Big) \partial_\alpha \dot{\xi}_l - (\dot{\xi}_r + \dot{\xi}_l)(\lambda+\eta) \Bigg] dt \\
& + (\mu_r + \dot{\lambda}) \partial_\alpha \xi_r(0) - \dot{\lambda} \partial_\alpha \xi_r(T) + (\nu_r - \lambda) \partial_\alpha \dot{\xi}_r(0) + \lambda \partial_\alpha \dot{\xi}_r(T) \\
& + (\mu_l + \dot{\eta}) \partial_\alpha \xi_l(0) - \dot{\eta} \partial_\alpha \xi_l(T) + (\nu_l - \eta) \partial_\alpha \dot{\xi}_l(0) + \eta \partial_\alpha \dot{\xi}_l(T) \qquad (7.78)
\end{aligned}$$

where the term $\mathcal{H}_f'|_{u_0} \approx u_L/u_0$ by linearization. Since the partial derivatives of the displacement ξ w.r.t. the model parameter α is difficult to compute, we cancel out the related terms by setting

For $0 < t < t_m$:

$$\ddot{\lambda} + \left(2\beta \xi_r \dot{\xi}_r + 1 - \frac{\Delta}{2} \right) \lambda + 2\tilde{c} dR \mathcal{H}_f' \Big|_{u_0} = 0 \qquad (7.79)$$

$$\ddot{\eta} + \left(2\beta \xi_l \dot{\xi}_l + 1 + \frac{\Delta}{2} \right) \eta + 2\tilde{c} dR \mathcal{H}_f' \Big|_{u_0} = 0 \qquad (7.80)$$

$$\beta(1+\xi_r^2)\lambda - \alpha(\lambda+\eta) = 0 \qquad (7.81)$$

$$\beta(1+\xi_l^2)\eta - \alpha(\lambda+\eta) = 0 \qquad (7.82)$$

$$(7.83)$$

with initial conditions

$$\text{At } t = t_m :$$

$$\lambda(t_m) = 0 \tag{7.84}$$

$$\dot{\lambda}(t_m) = 0 \tag{7.85}$$

$$\eta(t_m) = 0 \tag{7.86}$$

$$\dot{\eta}(t_m) = 0 \tag{7.87}$$

$$\tag{7.88}$$

Consequently, we obtain the derivative of F w.r.t. α

$$F_\alpha = \int_0^{t_m} -(\dot{\xi}_r + \dot{\xi}_l)(\lambda + \eta)dt \tag{7.89}$$

Similarly, we obtain the derivatives of F w.r.t. β and Δ

$$F_\beta = \int_0^{t_m} \left((1+\xi_r^2)\dot{\xi}_r\lambda + (1+\xi_l^2)\dot{\xi}_l\eta\right) dt \tag{7.90}$$

$$F_\Delta = \int_0^{t_m} \frac{1}{2}\left(\xi_l\eta - \xi_r\lambda\right) dt \tag{7.91}$$

7.2.4 Parameter Estimation Algorithm

The algorithm for solving the parameter estimation problem (7.27) is outlined below.

1. Integrate (7.21) and (7.22) with initial conditions (7.23), (7.24), (7.25) and (7.26) from 0 to t_m, obtaining ξ_r^k, ξ_l^k, $\dot{\xi}_r^k$ and $\dot{\xi}_l^k$.
2. Solve the forward propagation model (7.34) for u_L^k, $\mathcal{H}_f'\Big|_{u_0^k}$.
3. Calculate the estimation difference r^k using (7.42).
4. Solve the backward propagation model (7.54) for z^k.
5. Update f^k using (7.70).

6. Integrate (7.79), (7.80), (7.81) and (7.82) with initial conditions (7.84), (7.85), (7.86) and (7.87) from t_m to 0, obtaining λ^k, $\dot{\lambda}^k$, η^k and $\dot{\eta}^k$.

7. Update α, β and Δ with (7.74).

We adopt the simple gradient descent method. However, other gradient-based optimization approaches, such as the conjugate gradient method, can also be used.

7.2.5 Numerical Solution for Wave Propagation

Now what remains for us to do is to solve the acoustic wave propagation problems (7.34) and (7.54). We derive a finite element solution for them.

Variational Formulation

First, for the time-dependent system of PDEs, we discretize it along time t with the backward Euler method [9], yielding a sequence of differential equations. We split the time domain T into N uniform length intervals Δt. For time step n, $0 \leq n \leq N-1$, applying the backward Euler method to the left side of (7.28) gives

$$\left[D_t D_t^- u\right]^n \coloneqq D_t D_t^- \left(\frac{\partial^2 u}{\partial t^2}\right) = \frac{u^n - 2u^{n-1} + u^{n-2}}{\Delta t^2} \tag{7.92}$$

where $D_t D_t^n$ is a finite difference operator w.r.t. time at time step n [9, 10]. Substitution into (7.28) yields

$$\left[D_t D_t^- u = c^2 \frac{\partial^2 u}{\partial x^2} + f\right]^n \tag{7.93}$$

$$\Leftrightarrow \quad u^n = \Delta t^2 c^2 \frac{\partial^2 u^n}{\partial x^2} + \Delta t^2 f^n + 2u^{n-1} - u^{n-2} \tag{7.94}$$

Next, define the residual at time step n as

$$R^n = u^n - \Delta t^2 c^2 \frac{\partial^2 u^n}{\partial x^2} + \Delta t^2 f^n + 2u^{n-1} - u^{n-2} \tag{7.95}$$

Applying Galerkin's method [9, 11] gives

$$\langle R^n, v\rangle_{W^{k,2}} = 0 \tag{7.96}$$

where $v \in \mathcal{W}^{k,2}$ ($\mathcal{W}^{k,2}$ is the Sobolev space of functions with bounded L_2 norm and k-th order weak derivatives) is a qualified test function. Galerkin's method orthogonally projects the residual to the function space $\mathcal{W}^{k,2}$. Expanding (7.96) yields

$$\int_\Omega u^n v dx - \Delta t^2 c^2 \int_\Omega \frac{\partial^2 u^n}{\partial x^2} v dx = \int_\Omega \left(\Delta t^2 f^n + 2u^{n-1} - u^{n-2}\right) v dx \tag{7.97}$$

Integration by parts for the second-order term in (7.97) gives

$$\int_\Omega \frac{\partial^2 u^n}{\partial x^2} v dx = -\int_\Omega \frac{\partial u^n}{\partial x}\frac{\partial v}{\partial x} dx + \int_\Gamma \frac{\partial u^n}{\partial n_\Gamma} ds \tag{7.98}$$

where n_Γ is the outward normal unit vector of the boundary Γ, and ds is the 1-form [1] on Γ. For problem (7.34), applying the boundary condition (7.31) and substitution (7.98) back into (7.97) yields the variational problem

$$\int_\Omega u^n v dx + \Delta t^2 c^2 \int_\Omega \frac{\partial u^n}{\partial x}\frac{\partial v}{\partial x} dx = \int_\Omega \left(\Delta t^2 f^n + 2u^{n-1} - u^{n-2}\right) v dx \tag{7.99}$$

For problem (7.54), applying the boundary condition (7.51) and substitution (7.98) back into (7.97) yields a similar variational problem

$$\int_\Omega z^n w dx + \Delta t^2 c^2 \int_\Omega \frac{\partial z^n}{\partial x}\frac{\partial w}{\partial x} dx = \int_\Omega \left(\Delta t^2 f^n + 2z^{n-1} - z^{n-2}\right) w dx + \int_\Gamma r^n w ds \tag{7.100}$$

We can split the variational problem (7.99) into two parts

$$a(u, v) = \int_\Omega uv dx + \Delta t^2 c^2 \int_\Omega \frac{\partial u}{\partial x}\frac{\partial v}{\partial x} dx \tag{7.101}$$

$$L(v) = \int_\Omega \left(\Delta t^2 f^n + 2u^{n-1} - u^{n-2}\right) v dx \tag{7.102}$$

$$\tag{7.103}$$

where we have interchanged the unknown u^n with u. (7.101) is the bilinear form, and (7.102) is the linear form [9]. Our original problems (7.34) and (7.54) then reduce to solving

$$a(u, v) = L(v) \tag{7.104}$$

for each time step. By the Lax-Milgram Lemma [10], solving (7.104) is equivalent to solving the functional minimization problem

$$F(u) = \arg\min_{v \in \mathcal{V}} \frac{1}{2} a(v, v) - L(v)$$

By calculus of variations and taking the variation of the functional gives (7.104), hence the name variational form [9, 10].

Finite Element Approximation

For each time step, we solve (7.104) with the finite element method. We discretize the domain Ω with a mesh of uniformly spaced triangular cells. We take the P_2 elements as the basis function space, which contains piece-wise, second-order Lagrange polynomials defined over a cell. Each basis function has a degree-of-freedom (DoF) of 6 over a two-dimensional cell [9, 12]. Each element is associated with a coordinate map that transforms local coordinates to global coordinates and a DoF map that maps local DoF to global DoF [9, 12]. Each cell is essentially a simplex and can be continuously transformed into the physical domain.

Existence of Unique Solution The solution to the variational problem (7.104) exists and is unique [12].

Approximation Error The Galerkin's method gives the solution u_e with error bounded by $\mathcal{O}(h^3 \|D^2 u_e\|_{\mathcal{W}^{3,2}})$, where h is the cell size and D is the bounded derivative operator [10, 12].

Assume a solution $u = B + c^j \psi_j$ (using Einstein summation convention) with basis $\psi_j \in P_2$ and coefficients c^j. The function $B(x)$ incorporates the boundary condition

and, as an example, can take the form

$$B(x) = u_g + (u_m - u_g)\,\frac{x^p}{L^p}, \quad p > 0 \tag{7.105}$$

We also project $B(x)$ over the basis functions P_2 and express it as $B(x) = b^j\psi_j$. As a result, we obtain an unified expression $u = U^j\psi_j$ with U^j incorporating b^j and c^j. Similarly, we have $f^n = F_n^j\psi_j$, $u^{n-1} = U_{n-1}^j\psi_j$, $u^{n-2} = U_{n-2}^j\psi_j$. Set the test function as $v = \hat{\psi}_i$. Substitution into (7.101) and (7.102) yields

$$\begin{aligned} a(u,v) &= \int_\Omega U^j\psi_j\hat{\psi}_i dx + \Delta t^2 c^2 \int_\Omega U^j\psi_j'\psi_i' dx \\ &= \left(\int_\Omega \hat{\psi}_i\psi_j dx + \Delta t^2 c^2 \int_\Omega \psi_i'\psi_j' dx\right) U^j \qquad (7.106) \\ L(v) &= \int_\Omega \left(\Delta t^2 F_n^j\psi_j + 2U_{n-1}^j\psi_j - U_{n-2}^j\psi_j\right)\hat{\psi}_i dx \\ &= \Delta t^2\left(\int_\Omega \hat{\psi}_i\psi_j dx\right)F_n^j + 2\left(\int_\Omega \hat{\psi}_i\psi_j dx\right)U_{n-1}^j - \left(\int_\Omega \hat{\psi}_i\psi_j dx\right)U_{n-2}^j \qquad (7.107) \end{aligned}$$

Setting $M_{i,j} = \int_\Omega \hat{\psi}_i\psi_j dx$, $K_{i,j} = \int_\Omega \psi_i'\psi_j' dx$ and collecting (7.106) and (7.107) into matrix-vector form, we obtain

$$AU = b \tag{7.108}$$

where $A = M + \Delta t^2 c^2 K$, and $b = \Delta t^2 MF^n + 2MU^{n-1} - MU^{n-2}$. Hence, we reduce problem (7.104) into solving the linear system (7.108). Furthermore, the matrices M (known as the mass matrix) and K (known as the stiffness matrix) can be pre-calculated for efficiency.

7.2.6 Experiments

To show the validity of the proposed parameter estimation approach for the vocal fold-tract model, we continue with the experiments explained in the last chapter—estimating model parameters from clinically collected speech data and classifying voice pathologies. All experimental settings are the same as those presented in the last chapter. First, we obtain the same parameter estimation and pathology classification results as in Table 6.1. This shows that the ADLES method can accurately estimate

	Glottal Flow MAE		Parameter MAE	
	B-ADLES	FB-ADLES	α	Δ
Normal	0.021	0.022	0.042	0.049
Neoplasm	0.028	0.036	0.055	0.058
Phonotrauma	0.043	0.051	0.083	0.079
Vocal palsy	0.059	0.065	0.102	0.119
All	0.040	0.045	0.074	0.078

Table 7.2: Estimation error by backward and forward-backward approach.

model parameters. We can deduce voice pathologies by thresholding parameter ranges. Each parameter setting corresponds to a region in the bifurcation diagram (Figure 6-5) with characteristic phase space patterns, representing distinct vocal fold motions and thereby indicating different voice pathologies. Further, we compare the estimation precision in the proposed backward approach (previous chapter) and the forward-backward approach (this chapter). Table 7.2 shows the mean absolute error (MAE) of calculating glottal flows and parameters for four voice types (normal, neoplasm, phonotrauma, vocal palsy) obtained by the backward ADLES (B-ADLES) and forward-backward ADLES (FB-ADLES) procedures. The glottal flows obtained by inverse filtering the speech signals are treated as ground truths. Since there is no ground truth for the model parameters, we treat the parameters obtained by the backward ADLES (previous chapter) as ground truth. These results suggest that our forward-backward algorithm can effectively recover the vocal tract profile, glottal flow, and model parameters.

7.3 Neural Approaches for Solving PDEs

The finite element method is known for its efficiency, stability, and precision guarantee and is well-suited for solving large-scale PDE problems. Many well-established solvers exist in it, such as FEniCS [13], COMSOL [14], Ansys [15]. However, machine and deep learning advances also introduce practical approaches for solving PDEs. Compared to FEM approaches, neural networks are well-suited for solving PDEs due to their ability

to approximate highly nonlinear functions mesh-free, expressive power from over-parameterization, effective learning through large data and stochastic optimization, and efficient computation with hardware and software acceleration [16]. Neural nets are particularly attractive for high-dimensional PDEs, whereas conventional methods suffer from the curse of dimensionality [16, 17]. Neural nets with dynamical system constraints can better capture the underlying dynamics and make more physically plausible forecasts [18].

Physics-informed neural networks (PINNs) solve PDEs by using neural nets to approximate the PDE solution and minimize the estimation error (residual) at initial and boundary conditions and selected interior points [19]. The loss (error) functional can take strong or weak residual forms and other forms with constraints or regularization (e.g., boundary conditions, symmetries) [18, 20]. It differentiates the approximated solution in space and time to minimize the residual. PINNs have the advantage of dealing with strongly nonlinear systems and requiring a small amount of data [16]. The neural nets can be simple multi-layer perceptrons. PINNs work in continuous and discrete time regimes. They have demonstrated success in many fields, such as mechanics, fluid dynamics, and stochastic differential equations [16]. For instance, [21] proposed physics-constrained deep feedforward networks to solve the Navier–Stokes equations. It incorporates the governing PDEs into the loss while enforcing initial and boundary conditions, enabling training without simulated data.

Besides nonlinearity, another challenge in solving PDEs is high dimensionality: the solution complexity increases combinatorially as the degree-of-freedom increases [22]. A classic example is the Kolmogorov equation arising from stochastic processes in finance and stochastic optimal control problems [23]. Neural networks break the curse of dimensionality by approximating the solution with significantly reduced complexity. In the case of linear backward Kolmogorov PDEs, the complexity (the number of parameters) of the neural network solution increases at most polynomially with dimension [24]. The Feynman-Kac theorem states that the solution to the backward Kolmogorov equation is the conditional expectation of a stochastic process, which in turn is the solution of a stochastic differential equation (SDE) [25, 26]. Due to

the link between PDE and SDE, data can be generated to train the neural network solution by sampling the stochastic process as it evolves over time [22]. Both the estimation error and variance of the neural net solution decrease as the number of samples increases [16]. Similar neural nets and training approaches can be applied, as in PINNs. The Feynman-Kac formula also extends to general linear parabolic equations [27]. For solving variational problems, the admissible function class can be parameterized by neural nets, and the integrals in the energy functional can be evaluated by sampling [28].

Besides (deep) feedforward networks, more advanced network structures can be applied, such as convolutional neural nets, recurrent neural nets, autoregressive neural nets, and encoder-decoder networks [29, 30, 31, 32]. Having proven successful in natural language processing and computer vision, deep transformer-based architectures [33] may also be effective in solving PDEs. Dynamical systems also inspire some physics-guided network architectures, such as Turbulent-Flow Net [34] and the message-passing PDE solver with graph neural networks [29]. Particularly, the deep-rooted symmetry in physics leads to neural network designs with embedded invariance or equivariance [18]. However, neural approaches generally lack tools for systematic error analysis and deriving theoretical guarantees. Future work can be further focused on improving the precision of physics-guided neural PDE solvers and analyzing their theoretical properties.

7.4 Conclusions

This chapter extends the previous chapter by integrating the vocal tract into the phonation modeling dynamics. We present a forward-backward paradigm for effectively solving the coupled ODE-PDE system of the vocal fold-tract model. We also extend the ADLES method to solve the inverse problem of estimating the vocal fold-tract model parameters from observations and present an efficient numerical solution. An empirical study validates the proposed approach's utility in characterizing phonation dynamics and deducing voice pathologies. We also discuss important deep learning

approaches for solving PDEs. Future work can explore the integration of our approach with neural approaches.

References

[1] M. P. Do Carmo and J. Flaherty Francis. *Riemannian geometry*. Vol. 6. Springer, 1992.

[2] P. M. Morse and K. U. Ingard. *Theoretical acoustics*. Princeton university press, 1986.

[3] M. R. Portnoff. "A quasi-one-dimensional digital simulation for the time-varying vocal tract." PhD thesis. Massachusetts Institute of Technology, 1973.

[4] I. R. Titze and D. W. Martin. *Principles of voice production*. 1998.

[5] L. V. Kantorovich and G. P. Akilov. *Functional analysis*. Elsevier, 2016.

[6] K. Zhu. *Operator theory in function spaces*. 138. American Mathematical Soc., 2007.

[7] M. B. Giles and E. Süli. "Adjoint methods for PDEs: a posteriori error analysis and postprocessing by duality". In: *Acta numerica* 11 (2002), pp. 145–236.

[8] C. Dong and Y. Jin. "MIMO nonlinear ultrasonic tomography by propagation and backpropagation method". In: *IEEE transactions on image processing* 22.3 (2012), pp. 1056–1069.

[9] H. P. Langtangen and K.-A. Mardal. *Introduction to numerical methods for variational problems*. Vol. 21. Springer Nature, 2019.

[10] W. F. Ames. *Numerical methods for partial differential equations*. Academic press, 2014.

[11] V. Thomée. *Galerkin finite element methods for parabolic problems*. Vol. 1054. Springer, 1984.

[12] M. G. Larson and F. Bengzon. "The finite element method: theory, implementation, and practice". In: *Texts in Computational Science and Engineering* 10 (2010), pp. 23–44.

[13] M. Alnæs et al. "The FEniCS project version 1.5". In: *Archive of Numerical Software* 3.100 (2015).

[14] C. Multiphysics. "Introduction to comsol multiphysics®". In: *COMSOL Multiphysics, Burlington, MA, accessed Feb* 9.2018 (1998), p. 32.

[15] P. Kohnke. "Ansys". In: *Finite Element Systems*. Springer, 1982, pp. 19–25.

[16] J. Blechschmidt and O. G. Ernst. "Three ways to solve partial differential equations with neural networks—A review". In: *GAMM-Mitteilungen* 44.2 (2021), e202100006.

[17] M. A. Nabian and H. Meidani. "A deep learning solution approach for high-dimensional random differential equations". In: *Probabilistic Engineering Mechanics* 57 (2019), pp. 14–25.

[18] R. Wang and R. Yu. "Physics-guided deep learning for dynamical systems: A survey". In: *arXiv preprint arXiv:2107.01272* (2021).

[19] M. Raissi, P. Perdikaris, and G. E. Karniadakis. "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations". In: *Journal of Computational physics* 378 (2019), pp. 686–707.

[20] J. Berg and K. Nyström. "A unified deep artificial neural network approach to partial differential equations in complex geometries". In: *Neurocomputing* 317 (2018), pp. 28–41.

[21] L. Sun et al. "Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data". In: *Computer Methods in Applied Mechanics and Engineering* 361 (2020), p. 112732.

[22] C. Beck, A. Jentzen, et al. “Machine learning approximation algorithms for high-dimensional fully nonlinear partial differential equations and second-order backward stochastic differential equations”. In: *Journal of Nonlinear Science* 29.4 (2019), pp. 1563–1619.

[23] C. Beck et al. “Solving stochastic differential equations and Kolmogorov equations by means of deep learning. arXiv”. In: *arXiv preprint arXiv:1806.00421* (2018).

[24] A. Jentzen, D. Salimova, and T. Welti. “A proof that deep artificial neural networks overcome the curse of dimensionality in the numerical approximation of Kolmogorov partial differential equations with constant diffusion and nonlinear drift coefficients”. In: *Communications in Mathematical Sciences* 19.5 (2021), pp. 1167–1205.

[25] D. Revuz and M. Yor. *Continuous martingales and Brownian motion*. Vol. 293. Springer Science & Business Media, 2013.

[26] B. Øksendal. “Stochastic differential equations”. In: *Stochastic differential equations*. Springer, 2003, pp. 65–84.

[27] I. Karatzas and S. Shreve. *Brownian motion and stochastic calculus*. Vol. 113. Springer Science & Business Media, 2012.

[28] B. Yu et al. “The deep Ritz method: a deep learning-based numerical algorithm for solving variational problems”. In: *Communications in Mathematics and Statistics* 6.1 (2018), pp. 1–12.

[29] J. Brandstetter, D. E. Worrall, and M. Welling. “Message Passing Neural PDE Solvers”. In: *International Conference on Learning Representations*. 2021.

[30] Y. Bar-Sinai et al. “Learning data-driven discretizations for partial differential equations”. In: *Proceedings of the National Academy of Sciences* 116.31 (2019), pp. 15344–15349.

[31] J.-T. Hsieh et al. “Learning Neural PDE Solvers with Convergence Guarantees”. In: *International Conference on Learning Representations*. 2018.

[32] N. Wang, H. Chang, and D. Zhang. "Theory-guided auto-encoder for surrogate construction and inverse modeling". In: *Computer Methods in Applied Mechanics and Engineering* 385 (2021), p. 114037.

[33] A. Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[34] R. Wang et al. "Towards physics-informed deep learning for turbulent flow prediction". In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2020, pp. 1457–1466.

Chapter 8

Summary, Discussion, and Future Work

This chapter concludes this thesis by comparing the modeling approaches presented in previous chapters in a unified testing framework. Further, it summarizes and discusses the key results of this thesis and suggests some promising future directions that can be explored.

8.1 Comparing Models in a Unified Framework

We present a unified framework to compare the target-, data-, and process-specific models presented in the previous chapters. We select the task of age prediction from voice, which is a continuous-valued regression task. The specific task and data description is given in Section 3.2.3, Chapter 3. We have acquired the results for target-specific models from the following sources: Neural Regression Trees (NRT) in Section 3.2.3, Chapter 3, Hierarchical Routing Mixture of Experts (HRME) in Section 4.3.3, Chapter 4, and the results for data-specifc models from the sources: Class-Dependent Adversarial Latent Structure Matching (CALM) in Section 5.3.3, Chapter 5.

The process-specific phonation model—asymmetric vocal folds model and the Adjoint Least Squares (ADLES) parameter estimation method described in Section 6.3.3,

Chapter 6—is designed for classification tasks (voice pathology classification), and they do not produce fine-grained categories. To utilize the physical model for a regression task and predict the speaker's height, we adopt the approach proposed in [1] and [2]—deriving features from the phonation model and using them as input to a neural regression model. Specifically, for each speech signal in the Fisher dataset [3], we extract the `/i/` vowels using a state-of-the-art automatic speech recognition system. The vowel `/i/` carries more distinctive information than other vowels [1, 2]. These extracted sounds are segmented into 100 ms chunks, corresponding to 800 samples each, and segments less than 100 ms are discarded. We use the ADLES method for each segment to estimate the asymmetric vocal folds model parameters α and Δ and produce the estimated glottal flows (see Chapter 6). We compute the difference between the predicted and actual glottal flows (from inverse filtering) and obtain normalized difference vectors of fixed lengths. Each difference vector is concatenated with the estimated model parameters and the estimation residual, forming an augmented feature vector. These feature vectors are fed into a three-layer feedforward network to predict the speaker's age.

Table 8.1 shows the results for age estimation obtained by the three modeling categories. Our target-specific model HRME-MLP achieves the best performance. Although not directly modeled for predicting age, the data-specific approach CALM-MLP and process-specific approach ADLES-MLP also yield reasonably good performance, suggesting the features extracted from our CALM framework and phonation modeling process (i.e., the ADLES framework) are expressive enough and useful for various downstream target-specific tasks. Notably, the phonation behaviors are controlled by the articulation configurations of the speaker, which in turn are influenced by the speaker's physical and other profiling parameters (such as age). Hence, by recovering the phonation model parameters and the underlying physical dynamics, our ADLES framework is well-positioned to produce discriminative features for characterizing and profiling speakers.

Table 8.1: Age Estimation Results

Category	Method	Male		Female	
		MAE	RMSE	MAE	RMSE
Target-specific	SVR	9.22	12.03	8.75	11.35
	MLP	9.06	11.91	8.21	10.75
	NRT	7.20	9.02	6.81	8.53
	HRME-MLP	**6.91**	**8.74**	**6.40**	**8.07**
Data-specific	CALM-MLP	7.28	9.64	7.25	9.58
Process-specific	ADLES-MLP	8.46	11.05	7.96	10.21

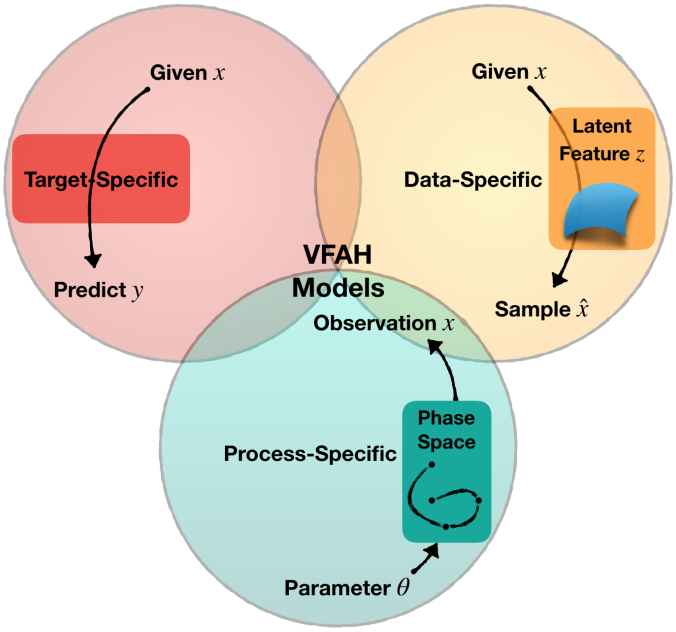


Figure 8-1: Diagram of the three model categories for VFAH.

8.2 Summary and Discussion

We conclude this thesis work with the same diagram shown at the beginning (Figure 8-1). This thesis extensively studies the modeling approaches for voice-based forensic analysis of humans (VFAH). VFAH aims to derive information from a person's voice to estimate or deduce the profile parameters of the speaker in a language-agnostic manner. Such profile parameters include physical traits (e.g., height, weight, facial skeletal contour), physiological traits (e.g., heart rate, blood pressure), psychological traits (e.g., mood, emotion, stress, mental disease), medical traits (e.g., disease, illness), demographic traits (e.g., age, gender, ethnicity, nationality, religion), sociological traits

(e.g., education, occupation, social status), and other bio-parametric or bio-descriptive traits. This thesis uses a subset of these profiling tasks as examples and focuses on them to highlight the strengths and weaknesses of different modeling approaches in this context. The chosen tasks are important ones, such as speaker identification, age and height estimation, gender and dialect classification, and voice disorder characterization. However, the methodologies and understanding developed in this thesis and the thesis's scope naturally extend to many other profiling tasks.

As mentioned above, this thesis studies a broad spectrum of voice processing and computational modeling methodologies, technologies, and tools for VFAH. We specifically investigate three broad modeling categories: (1) target-specific models, (2) data-specific models, and (3) process-specific models. We build theoretical formulations and practical algorithms and validate them with relevant, systematically conducted experiments.

Target-Specific Models Target-specific models have a specific analysis target of interest, aiming to derive information from the human voice and to make decisions about the analysis target. We develop feature representations for human voices that capture the most target-descriptive information and supervised machine and deep learning models to learn from these features and predict the target. We show that both strategies are effective and can be combined to give optimal results. Target-specific modeling has historically been the most widely used approach for such tasks.

Specifically, in the target-specific modeling context, we explore and analyze the potential of using breath sounds during inhalation for identifying speakers [4]. Intra-speech breath sounds carry unique information about the configurations of the various structures and geometry of the vocal tract, from the lungs and trachea to the larynx and oral and nasal cavities. These physical configurations are influenced by the speaker's profile parameters. Hence, intervocalic breath sounds embed characteristic signatures of the speaker. Further, breath sounds (especially intervocalic breath sounds) are ubiquitous, measurable, and invariant under disguise and impersonation, as they are usually not under the speaker's voluntary control and are extremely difficult to modify

consistently. This allows us to exploit the intervocalic breaths to reveal information about the speaker's identity. We present a constant-Q feature representation that effectively manifests the speaker-discriminatory resonance patterns of breath sounds and a CNN-LSTM framework that combines representation learning, speaker modeling, and decision making into a single pipeline.

We also study target-specific models for two challenging VFAH tasks: age and height estimation from voice. We bypass the difficulties of direct modeling approaches and recast the regression problem into a regression-via-classification (RvC) framework and present neural regression trees (NRT) [5] for such tasks. NRT adopts a divide-and-conquer strategy and develops a hierarchical partitioning policy to find the optimal discretization of the target variable based on each split's local optimality. In addition, NRT also optimizes the local features at each node to be more discriminative. We present an algorithm with a triviality loss to optimize partition boundaries, node classifiers and features jointly.

We extend the NRT model to deal with complicated data distributions and present a new modeling approach: the hierarchical routing mixture of experts (HRME) [6]. Addressing the difficulty of partitioning and routing data in conventional approaches and the sub-optimality of NRT, HRME introduces a novel gating mechanism that jointly partitions the input-output space based on the natural separability of the multimodal data and routes the data to simple regression models for reliable predictions. Furthermore, we formulate a probabilistic framework for HRME and construct an effective recursive Expectation-Maximization (EM) algorithm to jointly optimize the input-output partition, tree structure, and expert models.

Data-Specific Models Data-specific models aim to discover, extract, represent, and exploit the most intrinsic information in the human voice. They are not tied to a specific analysis target, but the derived feature representations are readily applicable to various profiling tasks. Such models are generative—they model the underlying data distribution and can generate samples from it. Generative models allow us to distill intrinsic data representations—latent features—from within their latent space. However,

the utility of these latent features is limited by their high dimensionality, inseparability, and unidentifiability. Addressing these issues, we present a class-dependent adversarial latent structure matching (CALM) framework. CALM automatically discovers low-dimensional latent features that are disentangled and geometrically separable. They disambiguate the influences of different profiling parameters on voice. Further, the separable latent space constructed by CALM is class-dependent, resulting in naturally clustered latent features directly usable for classification and regression tasks. In addition, CALM imposes an algebraic structure to the latent space that enables sampling and interpolation within/across classes. Such algebraic operations can be semantically interpreted to understand how the sample space continuously changes with the latent space. Consequently, CALM provides an effective tool for discovering and analyzing the relationship between latent features and the corresponding profile parameters.

Process-Specific Models Process-specific models are physical models that represent a specific physical process mathematically. These models describe the observed data through dynamical systems comprising ordinary or partial differential equations (ODEs/PDEs) with constraints. At the same time, the observation space is further dictated by the dynamics in the underlying phase space. This thesis studies one physical process of particular interest—the phonation process. Phonation is the process of producing voiced sounds. It involves a complex and delicate interplay of the physical articulatory instrument, aerodynamic forces across the glottis, and the cognitive and mental processes that influence voice production. By developing process-specific models, we can characterize such processes and reveal many fundamental traits of the speaker in a language-agnostic manner. Particularly, we present the asymmetric vocal folds model for phonation and the ADLES algorithm to accurately and efficiently recover the model parameters and solve the oscillatory dynamics of vocal folds [7]. The ADLES framework studies the dynamical system's behaviors through its phase space characteristics, such as stability and bifurcation. Such characterization provides a tool to analyze different phonation phenomena and physiological aspects of the human

voice. One example of the utility of this method is that we can deduce different voice abnormalities by the simple placement of entrainment behavior on the bifurcation map. We can also derive features from the dynamical system for various target-specific profiling tasks. Further, we extend the vocal folds model by incorporating the vocal tract and present a forward-backward paradigm for solving coupled ODE-PDE systems. We also extend the ADLES method to solve the inverse problem of estimating vocal fold-tract model parameters from observations.

Each of the three model categories has its strengths and weaknesses, and models from one category often cross the fluid boundary and aid the other. Together, they promote robust feature representations and effective modeling and algorithmic approaches that demonstrate success in various VFAH tasks. These models also give us valuable perspectives and insights into the forensic analysis of human voices.

8.3 Future Work

In target-specific modeling, we can explore the ability of other phonemes besides breath sounds to identify speakers and predict other profile traits. We can evaluate other feature representations and modeling choices, especially those robust to adverse conditions such as noise and disguise.

In data-specific modeling, we can further analyze the latent space's geometric, topological, and algebraic structures. We can improve the stability of the adversarial learning approach and explore other ways of imposing finer-grained latent structures. These latent structures can provide a direction for more interpretable representation learning. We can also try to improve the generalization and adaptability of latent features to downstream tasks.

In process-specific modeling, we can explore the utility of other physical models of voice production and other physical processes pertaining to VFAH. We can also improve the precision, stability, and efficiency of the estimation algorithms and numerical solutions and provide theoretical guarantees.

Another direction of suggested research is characterizing the phase space from

algebraic perspectives. The phase space characterization presented in this thesis is based on phase space trajectories (a topological perspective), which are local and volatile. We can find a global and more stable characterization of the phase space from an algebraic perspective. Algebraic invariants represent one of many feasible approaches. An *algebraic invariant* is an algebraic structure that is invariant under topological transforms, i.e., continuous deformations. Hence, we recast the study of the topological structures of the phase to the study of its algebraic constructs, such as homotopy groups and homology/cohomology groups, which are easier to classify. Such an algebraic approach can provide a systematic and refined deconstruction of the phase space. More generally, from a category theory perspective, algebraic invariants are *functors*—they describe not only homeomorphic topological spaces but also the morphisms (maps) between them. For example, algebraic invariants can characterize the homeomorphisms between phase spaces (e.g., evolution maps, poincaré maps) and reveal large-scale structures and global properties (e.g., existence and structure of orbits).

Lastly, as discussed in Chapter 6 and 7, we explore and build upon the deep connection between dynamical systems and deep neural models. On the one hand, we can study deep learning approaches for solving and analyzing dynamical systems. On the other hand, we can draw insights from the study of dynamical systems in interdisciplinary settings, connecting areas such as physics, nonlinear analysis, geometry, topological manifolds, algebra, and optimal control. Thus, we can explore the integration of dynamical systems and deep neural models, borrow from the well-established theories and tools in these fields to analyze and interpret the behaviors of deep neural models (such as studying the phase space structures of deep neural nets), and gain valuable insights and guidance to advance deep learning theories and applications in the context of human profiling from voice.

References

[1] M. Al Ismail, S. Deshmukh, and R. Singh. "Detection of COVID-19 through the analysis of vocal fold oscillations". In: *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2021, pp. 1035–1039.

[2] S. Deshmukh, M. Al Ismail, and R. Singh. "Interpreting glottal flow dynamics for detecting covid-19 from voice". In: *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2021, pp. 1055–1059.

[3] C. Cieri, D. Miller, and K. Walker. "The Fisher corpus: a resource for the next generations of speech-to-text." In: *LREC*. Vol. 4. 2004, pp. 69–71.

[4] W. Zhao, Y. Gao, and R. Singh. "Speaker identification from the sound of the human breath". In: *arXiv preprint arXiv:1712.00171* (2017).

[5] S. A. Memon et al. "Neural regression trees". In: *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2019, pp. 1–8.

[6] W. Zhao et al. "Hierarchical routing mixture of experts". In: *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE. 2021, pp. 7900–7906.

[7] W. Zhao and R. Singh. "Speech-based parameter estimation of an asymmetric vocal fold oscillation model and its application in discriminating vocal fold pathologies". In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2020, pp. 7344–7348.