DOMAIN ADAPTATION VIA ADVERSARIAL LEARNING FOR SPEECH EMOTION RECOGNITION

A THESIS SUBMITTED TO THE UNIVERSITY OF MANCHESTER
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
IN THE FACULTY OF SCIENCE AND ENGINEERING

2021

Hao Zhou

Department of Computer Science

Contents

List of Tables

List of Figures

Abstract

Speech emotion recognition plays an important role in creating more intelligent agents and systems. A lack of suitable speech emotion data, however, often occurs and hinders building the practical systems. In order to tackle this issue, knowledge transfer or domain adaptation, has emerged as a promising solution, which features leveraging a related information-rich source to help optimize the performance on the target task. In spite of the great progress of adversarial learning based domain adaptation techniques in computer vision, so far rare works have attempted to apply these advanced techniques on speech emotion recognition. This project explores whether and how adversarial learning can be used to eliminate the divergence or domain shift that exists in speech emotion data. We particularly address the scenario of supervised domain adaptation (SDA), where only very limited labelled data from the target domain are available. We propose Class-wise Adversarial Domain Adaptation (CADA) to reduce the domain shift for all common classes between the target and source domains via adversarial learning. Different from general practices, CADA combines the class discriminator and domain discriminator into one architecture, and the training algorithm is straightforward with either multi-layer perceptrons or deep neural networks as the basis of the model. We also extend CADA to the unsupervised scenario when only a few unlabelled target-domain data are available. We systematically estimate CADA with real-world speech emotion datasets under many different practical settings and demonstrate the effectiveness of CADA with an advantage over ordinary fine-tuning technique and the state-of-the-art adversarial-learning based domain adaptation approach.

Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright

Acknowledgements

It is definitely not easy to do a PhD, and finally, it is now close to the end. There are many things I will remember about this journey, this experience, this university, and this country.

I would like to thank my supervisor, Ke Chen, for his continuous support to my work, and also at the very beginning, providing me this precious opportunity to study in a great research group at a renowned university. I want to thank my colleagues, who have been very good examples to me, both in life and in study.

I want to thank my good friends in the UK, including Jing, Jiaxi, Xuli, Meng, Zhu, Dongjiao, Peizhi, Qian, Sophia, Fabio, Mike, Tudor, and Prae, as well as several very local friends, who have been so nice and inspiring to me.

At last I want to thank my parents, who are always so proud of me, my wife, a miracle in my life, and my baby daughter, a perfect gift in the pursuit of happiness.

I am lucky to be a student of University of Manchester and to study in the UK. I believe wherever I am in the future, I will always keep a touch with this warm place.

Chapter 1

Introduction

1.1 Motivation

A highly important role in human expression and communication, emotions refer to specific and intensive mental activities. Although scientists have not reached a consensus on the definition of emotions, there have developed many theories covering the origin, categorization, functions of emotions [117, 74]. In the past decades, artificial intelligence technologies have been leveraged to study emotions, yielding a new inter-disciplinary branch called affective computing [80], which is aimed to build more intelligent and human-like agents or systems that can detect, recognize, and interpret human emotions in the future. The applications of affective computing can be found in many real-world scenarios. For example, it has already been used for initial assessment of some psychological diseases such as depression [26], autism [98], and bipolar disorders [87]. It also shows a promising prospect in calling-centre service [61], intelligent automobile systems [49], and entertainment industry [71], etc.

The main task of affective computing is automatic emotion recognition (AER), which can utilise the information of speech, body gestures, or facial expressions. Particularly, speech is a fast, efficient, and essential communication manner between humans in daily life. It can be the only source of information when no pictures or videos of the speaker are available, e.g. in the scenario of the calling-service centre. As a result, speech emotion recognition (SER) [22] is one of the main research directions in affective computing. Generally, the information embedded in speech can be categorized as linguistic or paralinguistic. Most works regarding SER exploit paralinguistic information, from which many kinds of hand-crafted acoustic features can be extracted [97, 24].

Machine learning methods rely on data. Nevertheless, due to the special nature of speech emotion data, it is extremely difficult to acquire samples from natural conversations [22]. One reason is that collecting these data may lead to privacy violation, and furthermore, the emotional state in most speech in normal life is neutral and collecting the speech with desired emotions can be unrealistic. Consequently, the speech emotion data used by researchers are usually generated from laboratories under certain pre-designed conditions. While on one hand, an increasing number of high-quality emotional speech databases/corpora have become available [22, 96]. On the other hand, the divergence between different corpora often leads to the problem that the performance of a recognition system trained on one corpus can degenerate dramatically when tested on another corpus. It is straightforward to understand the occurrence of such a divergence, also named dataset shift or domain shift, as different speakers from different backgrounds under different recording environments can vary considerably in terms of emotion expression.

Transfer learning [78] or domain adaptation provides a solution to the issue of data scarcity in speech emotion recognition. The main idea of domain adaptation is that a related and information-rich source domain could be leveraged to help address the target task when the target domain suffers a lack of information. Specifically, the target domain may contain a very limited number of labelled data, making it difficult to establish a robust recognition system. Under this situation, some existing databases can be used to improve the performance on the target task via domain adaptation. This scenario, where the target domain contains only very few labelled examples, is termed supervised domain adaptation (SDA). The other scenario, unsupervised domain adaptation (UDA), refers to where the target domain has many examples but no label information is available.

In spite of some domain adaptation works in SER, very few of them are under the context of supervised domain adaptation. However, we believe SDA is a practical scenario for speech emotion recognition. For example, because of the high difficulty of collecting real-world speech emotion data, there may be very few examples representing natural emotions available for the target task. It would be desirable to use an existing large database containing acted emotion examples for better performance on the target task.

Domain adaptation can be achieved by different kinds of techniques for different applications [38, 78]. In recent years, a branch of domain adaptation methods featuring adversarial learning [32, 110, 27] have proven very successful for tasks regarding

computer vision. The state-of-the-art SDA method, namely few-shot adversarial domain adaptation (FADA) [70], generates data pairs by combining the source and target examples and then performs adversarial learning with these pairs to maximise domain confusion (i.e. minimise domain shift). Nevertheless, we find that FADA cannot work well for speech emotion recognition, and we observe that this is because high intra-class variability extensively occurs in speech emotion data while the pairing technique in FADA ignores specific class information in different domains. In order to tackle this issue, we propose Class-wise Adversarial Domain Adaptation (CADA) which is aimed at reducing the domain shift for all common classes between the source and target domains. CADA can be implemented based multi-layer perceptrons or deep neural networks with straightforward training algorithm. We provide a systematical evaluation on CADA with toy data and real-world datasets under different experiment settings. It is verified that CADA is an effective supervised domain adaptation approach and superior to the state-of-the-art FADA. We also extend CADA to the scenario of unsupervised domain adaptation and empirically proves its effectiveness.

1.2 Aim and objectives

We aim to seek an effective and robust supervised domain adaptation approach to solving the problem of data scarcity in speech emotion recognition. Our specific objectives include

- Analyzing the factors that cause domain shift in speech emotion data;
- Identifying the limitations of the existing domain adaptation approaches for speech emotion recognition;
- Putting forward a new domain adaptation approach featuring adversarial learning;
- Evaluating the proposed approach with designed toy data and real-world datasets under multiple practical settings.

1.3 Achievements

In the pursuit of the objectives listed above, we mainly make the following achievements:

- A practical scenario, supervised domain adaptation, where the target domain contains very limited labelled data, is systematically investigated for speech emotion recognition. Within our best knowledge, this is the first systematic work addressing this issue for speech emotion recognition.
- A novel supervised domain adaptation approach CADA is proposed to overcome the limitation of the state-of-the-art approach FADA. Comprehensive experiments with toy data and real-world datasets verify that CADA outperforms FADA in terms of recognition accuracy.

1.4 Thesis structure

The rest of the thesis is structured as follows.

- Chapter 2 presents the basic knowledge regarding speech emotion recognition, including the concept of emotions and speech, their relationship, the general emotion recognition framework, the commonly used speech features, and the machine learning models, multi-layer perceptrons and 1D convolutional neural networks, which are used in this project.
- Chapter 3 focuses on the speech emotion databases, including how they are generated and their characteristics. A summary of the databases used in this project is then presented.
- Chapter 4 discusses the related work about domain adaptation, including the concept of domain shift, traditional adaptation solutions, and latest methods featuring adversarial learning. The literature about applying domain adaptation to speech emotion recognition is also reviewed, and their relationship with our work is discussed.
- Chapter 5 introduces the proposed class-wise adversarial domain adaptation approach (CADA), including the basic idea, algorithm, model architecture, training algorithm, and examples illustrating the difference with other comparative approaches. The modified version of CADA for unsupervised domain adaptation, u-CADA, is also introduced.
- Chapter 6 shows the systematical evaluations based on multi-layer perceptrons on the CADA and u-CADA with real-world databases under different designed experiment settings.

- Chapter 7 explores using CADA for deep learning models (1D convolutional neural networks) and conducting relevant experiments.
- Chapter 8 gives a summary of this thesis and discuss the limitations as well as the future work directions.

Chapter 2

Background Knowledge

This chapter gives the background knowledge about speech emotion recognition. Specifically, Section 2.1 discusses the conception of emotion from the viewpoints of different fields, including its definition, function, and expression forms. Section 2.2 explains how speech is produced and how emotion is expressed in speech. Section 2.3 introduces the inter-disciplinary field, affective computing, which studies emotions from the perspective of computer science. Section 2.4 outlines the general speech emotion recognition framework. Section 2.5 summarizes the commonly used speech features for building speech emotion recognition systems. Section 2.6 gives more details of machine learning models (multi-layer perceptrons and 1D convolutional neural networks) which are used in this project.

2.1 Understanding emotions

Emotions are the main object of this research, but what are emotions? Intuitively, emotions are some kind of bodily sensations with varying degrees of pleasure or displeasure, brought on by neurophysiological changes [79]. However, it proves highly difficult to define emotions.

The Oxford dictionary describes emotions as a strong feeling deriving from one's circumstances, mood, or relationship with others. In daily life, there are many other words we use that are actually intertwined with the concept of emotion, such as feeling, affect, mood, temperament, personality, etc. Within the field of affective neuroscience [79], these similar concepts are discriminated. For example, feelings are understood as a subjective representations of emotions, affect describes the underlying affective experience of an emotion, and moods are diffuse affective states that generally last

longer but are less intensive than emotions [47]. Scherer's component process model (CMP) [91] provides a useful perspective to understand emotions. The five crucial elements of emotions identified in CMP are

- cognitive appraisal: the evaluation of events and objects
- bodily symptoms: the physiological component
- action tendencies: the motivational aspect
- expression: the facial and vocal expression
- feelings: the subjective experience

Emotional experience is generated when all of these processes become coordinated and synchronized in a short period of time.

The origin and functions of emotions are not fully revealed, but it is agreed that emotions have a great influence on the physiological, behavioural, and cognitive development on humans [72]. Some works discover that the original role of emotions may be to motivate adaptive behaviours in humans that would have contributed to passing on of genes [90].

Study on emotions can be traced back to ancient times in both Western and Eastern societies. In philosophy, emotions are thought of closely related with human nature [46]. The great biologist Darwin systematically researched emotions in [16] which provides many insights on how emotions are expressed in humans and animals.

Over the past decades, research on emotions has increased quickly, with the contributions from psychology, neuroscience, medicine, history, sociology, and computer science. In particular, affective computing [80] is the branch of the study of artificial intelligence that focuses on human emotions, and it usually adopts an operative approach that favors an intuitive and relatively vague explanation on emotions. The emotion theories popular in affective computing are reviewed in Section 2.3.

2.2 Emotional speech

Speech is one of the main channels to convey emotions. In essence, speech is the intentional modulation of air pressure in order to transmit a message [45]. Generally, the vocal communication between humans can be described by speech chain, which

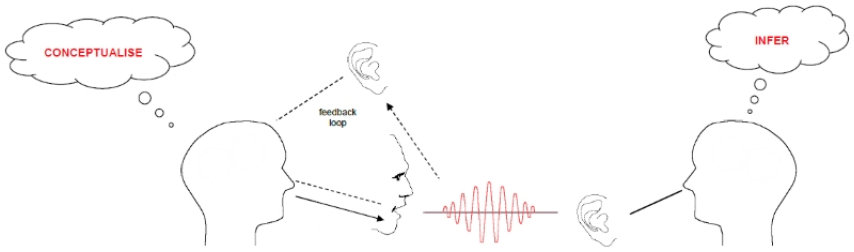


Figure 2.1: Speech chain

starts with forming an intention in the mind of the speaker and ends with understanding that intention in the mind of the listener.

As illustrated in Figure 2.1, speech is produced by human's vocal apparatus which is able to modulate air to produce sounds. This process starts in the lungs, crosses the trachea until the vocal tract, where the organs between the glottis, the vocal chords and the lips perturbs the air in order to produce sounds. Through the coordinated use of all these organs, humans are able to produce sounds whose fundamental frequency lies in the range of approximately 80-200 Hz, for males, and 180-400 Hz, for females [69] and to generate a variation of pressure of approximately 0.01-1 Pa at 1 meter from their lips [113]. Speech is perceived by the auditory system, a set of organs which transduces sounds and relays an electric signal to the brain. Sounds are collected by the outer ear, filtered by the membranes and the ossicles in the middle ear and finally converted into electric impulses in the inner ear. The auditory system in the human ear can perceive sounds between 20 Hz and 20 KHz; at different frequencies there are different minimal required intensities for a sound to be heard [116].

Because the variation of air pressure determined by speech can be described as a waveform that represents the change of pressure with time, speech can be treated as a time series signal, and common signal processing techniques are applicable to speech signals [76].

The feasibility of speech emotion recognition is based on the fact that various changes in the nervous system can indirectly alter a person's speech. For example, speech produced in an emotional state of fear, anger, or joy tends to be fast and loud with a wider range in pitch. However, in an emotional state of tiredness, boredom, or sadness, speech is often slow, low-pitched, and slurred [7]. This reveals what kind of information can be extracted as discriminative features to recognize emotions.

Information contained in speech signals include both the explicit semantic and other informative implicit contents. These two kinds of information can be regarded to

exist, respectively, in the linguistic layer and para-linguistic layer of a speech signal. In other words, a speech signal can be decomposed in two layers:

- Linguistic layer carries the semantic content which is made up by sounds constituting words. An effective communication requires the composition of words follow the common language rules reached by the speaker and listener. Speech recognition is the specific branch to address this semantic content.
- Para-linguistic layer carries information about the intentions and feelings of the speaker through acoustic cues, e.g. stresses and pauses. It also carries the information about the speaker himself/herself, through other acoustic cues such as tone and pitch, which can help the listener have a rough understanding of the speaker's gender, age, and other characteristics. Speech emotion recognition and speaker recognition mainly utilise these two kinds of information.

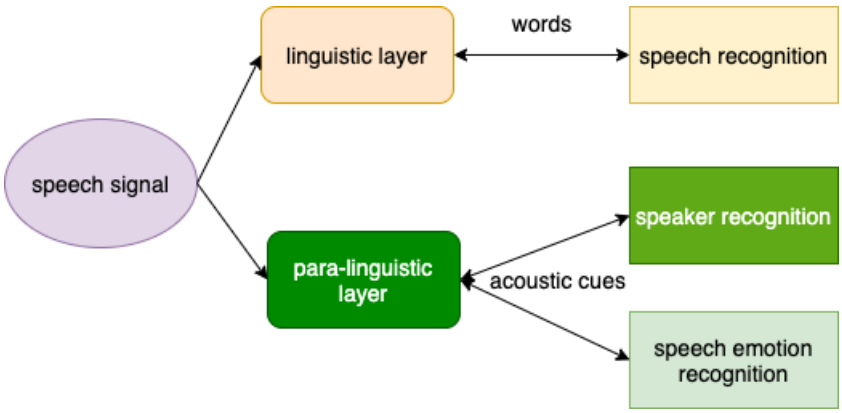


Figure 2.2: Decomposition of speech signals.

It should be noted that this layer-wise arrangement of speech is not a rigid decomposition. Some researchers regard the non-linguistic information as in the extra-linguistic layer (conveying the information about the speaker and background), or in the para-linguistic layer (conveying the information about emotions). In either way of decomposition, the boundaries between the layers are fuzzy as acoustic cues belong to more than one layer. In fact, study on speech has formed into three different directions in affective computing, as shown in Figure 2.2:

- Speech recognition [84, 42] aims to reconstruct the words uttered by the speaker;
- Speaker recognition [54, 29] attempts to identify the speaker through the analysis of an acoustic signal

- Speech emotion recognition [22, 55] intends to determine the emotional state of the speaker through acoustic cues.

It is evident that speech recognition utilises the linguistic information, while speaker recognition and speech emotion recognition usually rely on the para-linguistic information (or with linguistic information).

2.3 Affective computing

2.3.1 Overview

Affective computing is the study and development of systems that can recognize, interpret, and even simulate emotions. This interdisciplinary field spanning computer science, psychology, and cognitive science is usually believed to originate with Picard's work [80]. In practice, affective computing focuses on detecting emotional information, recognizing emotion classes, and simulating emotions.

- Detecting emotional information is about acquisition of emotional data via passive sensors, e.g. video cameras or microphones that are able to capture the data about the user's physical states or speech [28].

- Recognizing emotions depends on the extraction of meaningful patterns from the gathered data using machine learning techniques [2, 22]. As the name suggests, the goal of emotion recognition is to produce emotion class labels (predictions) that would match humans' predictions.

- Simulating emotions is an advanced objective of affective computing, in order to facilitate the interactivity between humans and machines or to create more intelligent agents [40].

Affective computing can be applied to many areas. Below shows some examples of the applications:

- Education [58]. Learners' development can be influenced by their emotional states. Affective computing technology can help teachers judge the learners' state so they can adjust their teaching plans. In distance education, there is usually no emotional incentive between teachers and students. Under such circumstances, students can easily get bored and distracted. Applying affective computing in distance education system can help solve this issue.

- Health care [115, 81]. Robots in health care equipped with affective computing technologies can better judge patients' emotional states, and accordingly, alter their actions or programming. Affective computing is also being applied to the development of communicative technologies for use by people with autism [93, 53].

- Entertainment [30]. Affective video games can access their players' emotional states through biofeedback devices. A particularly simple form of biofeedback is available through game pads that measure the pressure with which a button is pressed: this has been shown to correlate strongly with the players' level of arousal [103]. Affective games have been used in medical research to support the emotional development of autistic children [53].

- Social monitoring. There are many forms of social monitoring that can benefit from affective computing. For example, a car can monitor the emotion of all occupants and engage in additional safety measures, such as alerting other vehicles if it detects the driver to be angry [104] or shows fatigue [49]. An emotion monitoring agent can send a warning before one sends an angry email. Music players can select tracks based on mood. Companies use analysis about clients' facial expression to infer the respective market [31]. Calling-service centre can assess emotion states of the users [61] and improve their service quality.

2.3.2 Emotion theories

A preliminary question facing affective computing is how to define emotions that can be processed by computers. It proves that computational theories of emotions [68, 14], which have developed from psychology and cognitive science, match the goal of affective computing. These theories include categorical theories, continuous theories, and appraisal theories.

Categorical emotion theories

Categorical or discrete emotion theories postulate that emotions are discrete, measurable, and physiologically distinct. One of the most influential categorical theories is Ekman's Big Six [21], which classifies emotional states as anger, disgust, fear, happiness, sadness and surprise. Plutchik developed the 'wheel of emotions' [36], suggesting four groups of opposite emotions: joy versus sadness, anger versus fear, trust

Table 2.1: Common categorical emotions for speech emotion recognition. The first row lists Ekman's Big Six [21], one of the most influential categorical emotion theory.

anger	fear	sadness	happiness	surprise	disgust
pleasure	amusement	satisfaction	excitement	pride	shame
relief	guilt	fright	anxiety	jealousy	love
compassion	curious	bored	interested	relaxed	confident
affectionate	disappointed	worried	frustrated	contempt	aesthetic

versus disgust, and surprise versus anticipation. Some basic emotions can be modified to form complex emotions. Common categorical emotions are summarised in Table 2.1. Clearly, the majority of categorical emotions are negative. This observation is in line with our daily experiences that negative emotions can be of high diversity and complexity in contrast to positive emotions.

From a practical viewpoint, categorical theories are easy to use and to interpret, therefore they are widely applied in affective computing. On the other hand, they are also criticized for disregarding the fact that emotions are always culture-dependent and observer-dependent, and they may co-exist and mix in a complex way. Researchers have built some emotion corpora based on discrete theories, but these corpora vary considerably across the adopted emotion classes, posing a challenge when using different corpora [95, 63].

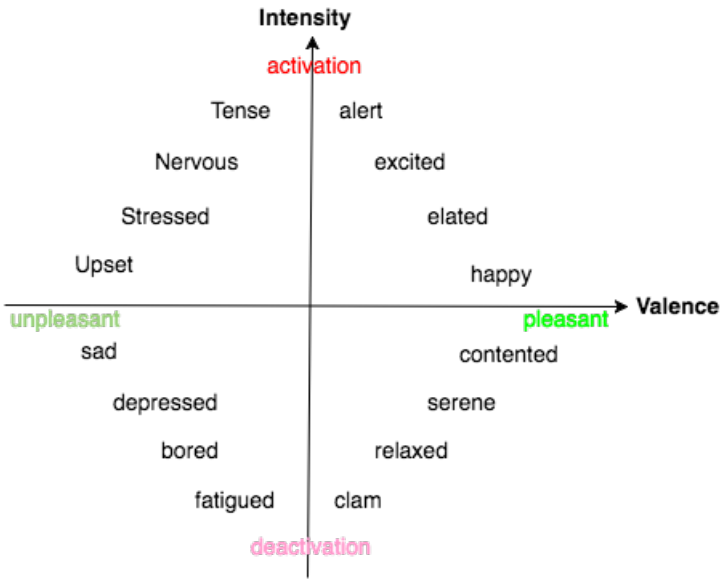


Figure 2.3: Emotional space defined by valence and intensity

Continuous emotion theories

Continuous emotion theories or dimensional theories [15, 33] postulate that emotions can be described by continuous dimensions representing different emotional fundamental properties. With these dimensions, all emotional states can be mapped to a multiple dimensional space. Two commonly-used dimensions are valence, measuring how negative or positive the experience feels, and intensity (or arousal), describing how strong the experience feels. They form a 2-dimensional space and different emotions fall into different areas in the space, as shown in Figure 2.3. Another popular dimension is dominance, which represents how dominant or controlling (versus controlled or submissive) the emotion is. Continuous theories of emotion have a high degree of versatility allowing researchers to consider many emotional states, especially non-prototype emotions. Because of these advantages, they are widely supported in affective computing. The difficulty using these theories is that the emotions can be hard to annotate.

Appraisal theories

Appraisal theories of emotion postulate that cognition and emotion are strictly interrelated. An emotion is, therefore, the result of a rational, though unconscious and often unexpressed, process evaluating a given situation. The complex process of evaluating events and circumstances can be subdivided in a collection of simpler judgements, each one considering only a particular feature of the event. These features are usually called appraisal variables or situational meaning structures [107]. The most important appraisal variables are novelty, pleasantness, goals, agency, norms [112, 52]. Like continuous theories of emotion, appraisal theories have a high degree of versatility and a broad support from the community combined with a strong explanatory power. However they also lack an implicit linguistic description and their annotation and evaluation is often considered too subjective.

2.4 Speech emotion recognition framework

To illustrate how speech emotion recognition works, a general framework is shown in Figure 2.4, with the raw audio signals as input and classification labels as output. It should be noted that all steps are not necessary in practice, and each step involves many options of operation that should be selected in use.

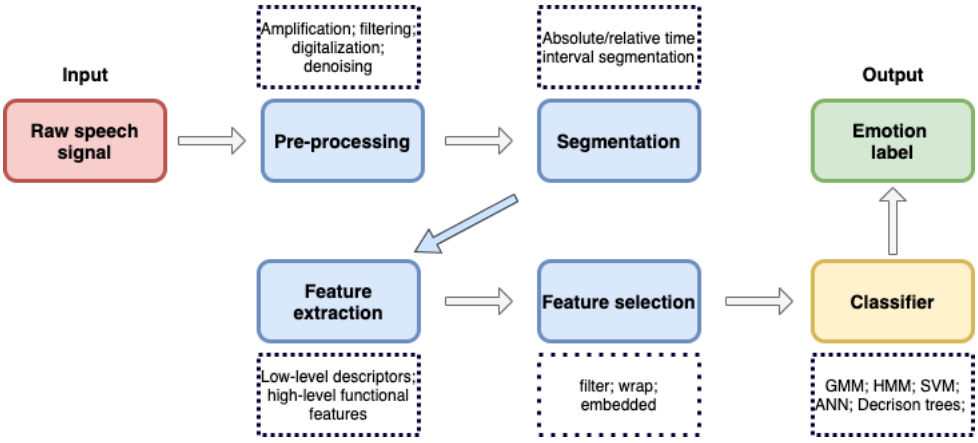


Figure 2.4: The general speech emotion recognition framework

2.4.1 Raw speech signals

Raw speech signals depicting the variation of amplitude with time are usually in the form of audio clips. The general form of an analogical continuous signal is $x(t) = f(t)$ where f is the function of amplitude. It is convenient to use periodic functions to describe speech signals and Fourier analysis is often adopted to convert the time-domain signals to frequency-domain signals. Typically, for continuous time-domain signal,

$$x(t) = \sum_{k=-\infty}^{\infty} a_k cos(2\pi k f_0 t) + b_k sin(2\pi k f_0 t) = \sum_{k=-\infty}^{\infty} c_k sin(2\pi k f_0 t + \phi_k) \tag{2.1}$$

where f_0 is the fundamental frequency, and a_k, b_k, c_k as well as ϕ_k are Fourier coefficients. There is corresponding transformation for discrete digital signals.

Emotional speech signals represented in the time-domain can be easily visualized using two-dimensional plots in which x-axis is time and y-axis is amplitude; this representation is useful to show the variation (intensity) and the timing of an emotional speech signal.

2.4.2 Pre-processing

Pre-processing is usually necessary to facilitate the processing of speech signals on a digital computer. The common technologies include:

- Amplification. This can increase the amplitude of a signal so that it is easier to tackle the signals of small intensity.

- Digitization. The raw analogue speech signal has to be digitized to be processed on a computer. This analogue-to-digital conversion (ADC) can be realized by sampling in time and sampling in amplitude. Sampling in time works by selecting a frequency or time step at which the continuous speech signal is sampled. Sampling in amplitude chooses a number of bits to represent the intensity of the signal.

- Filtering. This is to remove useless frequencies, misleading information, or noise. Depending on the nature of target tasks, different filters can be used such as high-pass filters, low-pass filters, band-pass filters, and notch filters.

- Denoising. This is the process that removes or reduces noise in the signal.

- Dereverberation. This is aimed to weaken the physical phenomenon that sound waves are reflected by solid objects back on their acoustic path with decreased amplitude (i.e. reverberation).

- Normalization or standardization. This is a general-purpose technique which regularizes the signal by mapping all values on a fixed scale.

The above provides an overview of the common pre-processing techniques in signal processing. In practice, what to use, how to use, and in which order to use are quite flexible and problem-specific.

2.4.3 Segmentation

As speech signals are usually not stationary, it is common to divide a speech signal into small segments called frames. Within each frame the signal is considered approximately stationary [85]. The ideal length of a frame may vary across applications, but it should be long enough to contain information related to emotions, but not too long so as to be unstable or non-stationary.

There have developed two strategies on segmentation of speech signals: linguistically-aware and linguistic-agnostic segmentation. As the name indicates, linguistically-aware segmentation relies on the semantic content and needs speech recognition module to segment the emotional speech into linguistic units such as words and phrases. Segmenting speech signals based the underlying phonemes [62] can also be regarded as belonging to this category. This approach relies on the observation that the spectral shapes of the same phoneme under different emotions are various (essentially true

for vowel sounds). However, the phoneme segmentation algorithm cannot produce a satisfactory performance.

On the contrast, linguistically-agnostic segmentation is unaware of the linguistic structure but utilises the information embedded in the paralinguistic layer of speech. Therefore this is the strategy mostly used for speech emotion recognition. Specifically, based on whether the frame is fixed in the length, there are two techniques as follows:

- Absolute time intervals (ATI) segmentation generates frames of equal length. It runs fast and the resulted frames are easily processed because of the uniform length. The optimal length is usually chosen empirically and it should guarantee the frames contain emotional information.

- Relative time intervals (RTI) segmentation extracts a fixed number of frames from an utterance and the frames may vary in length.

These two techniques can be combined and applied multiple times with different parameters.

2.4.4 Feature extraction

As a key issue in designing speech emotion recognition systems, feature extraction aims to find a suitable set of features that effectively characterize emotions.

There are many hand-crafted features that have been investigated for speech emotion recognition [97, 24, 22]. These features can further be categorised as local features and global features. While local features or low-level descriptors (LLDs) refer to those extracted from each frame, global features or functionals refer to the statistical features, e.g. the mean and standard deviation, based on the frames within an utterance.

In addition to hand-crafted features, which will be reviewed in details in Section 2.5, learned features [108] by deep models are attracting much interest nowadays thanks to the development of deep learning. However, the hand-crafted features show an advantage over learned features in terms of

- Interpretability. Hand-crafted features have explicit physical meanings from the perspective of signal processing, while learned features are generated via black box.

- Flexibility. Hand-crafted features can be flexibly selected and combined to suit different applications or models.

On the other hand, the hand-crafted features have some drawbacks:

- There are many groups or families of hand-crafted features, and it depends heavily on the experts' knowledge or experience on how to select and use them for optimal performance.
- The amount of hand-crafted features can increase dramatically when more low-level descriptors or functionals are added.

2.4.5 Feature selection

Feature selection is a basic skill in machine learning that can not only alleviate the curse of dimensionality but also improve the interpretability of the constructed models. Although for a given task, using more features normally results in better performance (assuming the data for training is sufficient), it is problematic to use as many as hand-crafted features. First, the number of acoustic features available for speech emotion recognition can be extremely huge, causing the curse of dimensionality. Second, the economical aspect should be considered in reality, and there must be a balance between the performance and the cost needed. Feature selection can be used to save the cost without significantly compromising the performance. In spite of many feature selection techniques in machine learning [35, 13], expert knowledge is very helpful in selecting the most suitable features for the target task.

2.4.6 Modelling

Many classic machine learning models have been investigated in speech emotion recognition. Currently, the most frequently used classifiers are linear discriminant classifiers (LDC), k-nearest neighbor (k-NN), Gaussian mixture model (GMM) [56], support vector machines (SVM) [1], artificial neural networks (ANN), decision tree algorithms and hidden Markov models (HMMs) [65, 73]. Various studies have showed that choosing the appropriate classifier can significantly enhance the overall performance of the system [22].

With respective advantages and disadvantages, it is difficult to decide which model works best for one task. The trend in recent years is on deep learning models [108, 25] which are able to learn hierarchical features. In particular, long short-term memory (LSTM) networks are often employed for continuous emotion estimation because of their ability to capture temporal context in speech signals [118].

2.5 Speech features

Speech features play a key role in building a robust recognition system. Many works have been carried out to find and design the informative features and estimate the effect. In addition to traditional acoustic features, recently, learned features are also tested in deep learning models. The deep models can even use processed raw audio signals as inputs, yielding impressive recognition accuracy [118].

2.5.1 Hand-crafted features

The relevance of different groups of features for speech emotion recognition is investigated in [97]. It also finds that using all groups can generally obtain better results than single groups, and there is not a fixed winner for all problems.

Low-level descriptors (LLDs) or local features denote those features extracted from individual frames. The commonly-used LLDs can be roughly grouped into the following families:

1. Prosodic features describe the prosodic phenomena including intonation, stress, rhythm, and voice quality. For example, intonation can be measured by fundamental frequency and contours. Stress or accentuation can be measured by energy, rhythm by duration and zero-crossing rate, and voice quality by spectral shape.

2. Spectral features describe the spectral nature, including formants, spectral roll-off (measure of the steepness of a transition in the frequency domain), spectral centroid, and spectral flux, etc.

3. Cepstral features describe the cepstral nature, including the value of cepstal coefficients, Mel-frequency cepstral coefficient (MFCC), and Mel Filter bank. In particular, MFCC is one of the most frequently used features in audio processing.

Global features or functionals refer to the statistical features based on local features. In other words, they synthesize the information embedded in the local features (given a fixed number of frames) in a single value. The common functionals are

1. Extreme functionals, e.g. the minimum and maximum.

2. Percentile functionals, e.g. the values of upper and lower quartiles.

3. Mean functionals, e.g. the arithmetic mean and the centroid.

4. Higher statistical moment functionals, e.g. standard deviation, variance, skewness, and kurtosis.

5. Other functionals, including statistical operators, ratio, error measures, linear or quadratic regression coefficients, and so on.

2.5.2 Learned features

It is a trend to build end-to-end speech emotion recognition models without using traditional hand-crated features [108]. The features, often learned by deep learning models, are thus called deep features or learned features. Learned features are usually abstract and informative representations, with poor interpretability in comparison to hand-crafted features. Besides, learning these features requires powerful computing capability. However, deep models without using traditional hand-crafted features can achieve a remarkable performance on some tasks.

2.5.3 The feature set GeMAPS

In order to achieve a standard estimation on the recognition performances for some contests, different fixed feature sets have been created by the organizers of those contests. For instance, the feature set InterSpeech09 is designed for the contest InterSpeech in 2009, which contains 384 features including 12 statistical functionals and 16 low-level descriptors (including additional delta coefficients).

In 2016, a number of top scientists propose the Geneva minimalistic acoustic parameter set (GeMAPS) [24] for voice research and affective computing. Extensive experiments demonstrate that GeMAPS and its extended version eGeMAPS [24] are comparable to the brute-force large scale feature sets. This motivates us to choose GeMAPS as the basic feature set in our work.

Specifically, GeMAPS is composed of 18 low-level descriptors (LLD), including frequency related parameters, energy related parameters, and spectral parameters. Details of these 18 LLDs are given in Table 2.2. All LLDs are smoothed over time with a symmetric moving average filter 3 frames long, and arithmetic mean and co-efficient of variation are applied as functionals, yielding 36 features. To loudness and pitch, 8 functions are additionally applied, which are 20-th, 50-th, and 80-th percentile, and the range of 20-th to 80-th percentile, and the mean and standard deviation of the slope of rising/falling signal parts. This gives a total of 52 features. Then the arithmetic mean of the Alpha ratio, the Hammarberg index, and spectral slopes from 0-500Hz

and 500-1500Hz over all unvoiced segments are included. Finally 6 temporal features are included: the rate of loudness peaks, the mean length and the standard deviation of continuously voiced regions, the mean length and the standard deviation of unvoiced regions, and the number of continuous voiced regions per second. In total, 62 features are contained in GeMAPS, and extra 26 features are added into GeMAPS generating its extended version, eGeMAPS.

2.6 Machine learning models

Regular machine learning models for speech emotion recognition have been briefly reviewed in Section 2.4. Here we give more details on two specific models, which, from the family of artificial neural networks, form the basis of the proposed approach in this project.

Artificial neural networks or simply called neural networks are computing systems inspired by the biological neural networks in human brains. ANNs are composed of artificial neurons, each having inputs and producing a single output which can be sent to multiple other neurons. A neuron is illustrated in Figure 2.5. Given the vector of input x, the output of the neuron is

$$f(x) = \delta(\sum_{i}^{n} w_i x_i + b)$$

where w is the vector of weight, b is the bias, and δ is the activation function. These

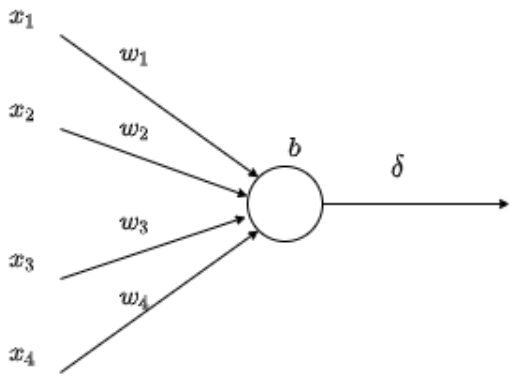


Figure 2.5: A neuron unit

three variables have important meanings, as w represents the strength of the connection

Table 2.2: GeMAPS feature set

Group	**Feature name**	**Description**
Frequency related parameters	Pitch	logarithmic F_0 on a semitone frequency scale
	Jitter	deviations in individual consecutive F_0 period lengths
	Formant 1,2,3 frequency	central frequency of first, second, and third formant
	Formant 1	bandwidth of first formant
Energy/Amplitude related parameters	Shimmer	difference of the peak amplitudes of consecutive F_0 periods
	Loudness	estimate of perceived signal intensity from an auditory spectrum
	Harmonics-to-noise ratio	relation of energy in harmonic components to energy in noise-like components
Spectral parameters	Alpha ratio	ratio of the summed energy from 50-1000 Hz and 1-5 kHz
	Hammarberg index	ratio of the strongest energy peak in the 0-2 kHz region to the strongest peak in the 2-5 kHz region
	Spectral slope 0-500 Hz and 500-1500 Hz	linear regression slope of the logarithmic power spectrum within the two given bands
	Formant 1,2,3 relative energy	ratio of the energy of the spectral harmonic peak at the first, second, third formant's centre frequency to the energy of the spectral peak at F_0
	Harmonic difference H1-H2	ratio of energy of the first F_0 harmonic to the energy of the second F_0 harmonic
	Harmonic difference H1-H3	ratio of energy of the first F_0 harmonic to the energy of the highest harmonic in the third formant range

between neurons, b is to ensure that the output value calculated through the input cannot be activated casually (i.e., working as a threshold). Activation function plays the role of non-linear mapping, which can limit the output amplitude of the neuron within a certain range, e..g [-1 1] or [0 1]. The most commonly used activation functions is the Sigmoid function, which can map the number of $(-\infty, +\infty)$ to the range of [0 1]. Other common activation functions include tanh, relu, and elu [101, 99]. Which activation function to use depends on the specific situation as each of them has unique characteristics.

Multi-layer perceptrons (MLPs) are classic neural networks, one of the most frequently used machine learning models. Convolutional neural networks (CNNs) are typical deep leaning networks that have demonstrated excellent performance in computer vision. These two types of neural networks will form the basis of the proposed method in this project.

2.6.1 Multi-layer perceptrons

A typical multi-layer perceptron (MLP) consists of three types of layers: input layer, hidden layer and output layer, as illustrated in Figure 2.6. The number of hidden layers cannot be too large as its learning algorithm, back-propagation, can be less effective with more layers added. This issue, known as vanishing gradient, has been studied in many works and can be avoided or alleviated through certain techniques [43, 44, 89].

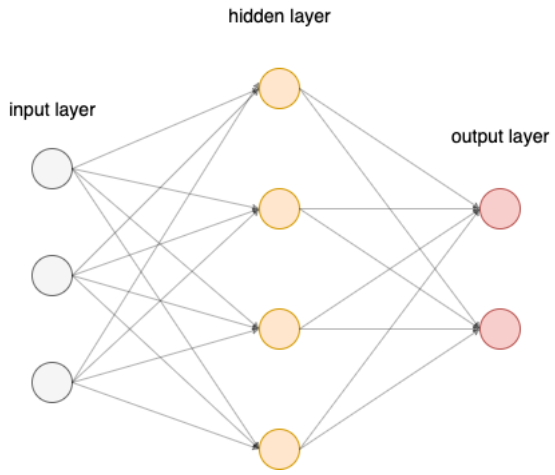


Figure 2.6: Architecture of MLP

The different layers of an MLP are fully connected, meaning that any neuron in the upper layer is connected to all neurons in the lower layer. Learning occurs in

the perceptron by changing connection weights after each piece of data is processed, based on the amount of error in the output compared to the expected result. This is an example of supervised learning, and is carried out through back-propagation [39], a generalization of the least mean squares algorithm in the linear perceptron.

2.6.2 Convolutional neural networks

Convolutional neural networks (CNN) [75] is an efficient method that has attracted widespread attention. It has become one of the research hot spots in many scientific fields, especially those regarding pattern classification.

CNN can recognize simple patterns in data and use them to form more complex patterns in higher layers. While 2-dimensional CNN is highly popular in computer vision, 1-dimensional CNN (1D CNN) can be very effective to obtain interesting features from a shorter (fixed-length) segment of the overall data set, and thus ideal for analyzing time series data such as audio signals. Another application of 1D CNN is natural language processing (NLP) [64], although LSTM networks [67] are more promising recently for NLP.

The basic architecture of 1D CNN is illustrated in Figure 2.7. It is characterized by using convolutional layer and pooling layer to extract feature representations from raw 1D input before feeding them to fully-connected layers for classification.

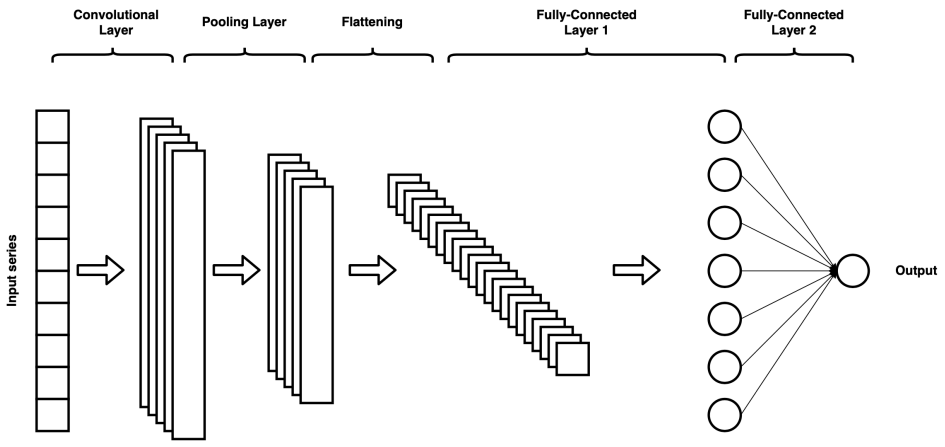


Figure 2.7: Architecture of 1D CNN

Specifically, the convolutional layer consists of a set of learnable filters (or kernels), which have a small receptive field, and extend through the full length of the input volume. During the forward pass, each filter is convolved across the length of the input

volume, computing the dot product between the filter entries and the input. Stacking the activation maps for all filters forms the full output volume of the convolution layer (as seen Figure 2.7). The pooling layer realises non-linear down-sampling for the output of the convolutional layer. There are several non-linear functions to implement pooling, where max pooling is the most common. It partitions the input into a set of pieces, and for each such sub-region, outputs the maximum. In practice, there can be several convolutional and pooling layers (they usually have equal numbers), although in Figure 2.7 only one convolutional layer and pooling layer are illustrated. Flattening is the next step that converts the output of pooling layers to 1D representations. The final classification is done via fully-connected layers. Neurons in a fully-connected layer have connections to all activations in the previous layer (in Figure 2.7, two fully-connected layers are shown).

2.7 Summary

This chapter presents the basic knowledge about emotions, emotional speech, speech emotion recognition and the machine learning models (neural networks) that are used in this project. In summary, speech is a kind of time-series signal containing the information about the speaker's emotional state. Such information, linguistic or non-linguistic, can be extracted and converted to acoustic features that suit machine learning models which are able to recognize the emotion classes.

Chapter 3

Speech Emotion Databases

This chapter explains how speech emotion databases/corpora are generally created, and how they are selected in this project (Section 3.1). Then the detailed information of the used corpora is presented in Section 3.2.

3.1 Building emotion databases

3.1.1 Emotion eliciting ways

Considering the complexity and vagueness of the definition of emotions, it is impossible to produce a database containing completely representative emotional speech. It is even challenging to establish a uniform database for comparison of different works in affective computing, due to a variety of issues about the data, such as languages and annotation methods. Many databases or corpora are application-dependent and they adopt different theories of emotion [50, 5, 34, 37, 102]. With respect to how the emotions are generated, there are three eliciting ways: acted, natural, and induced [22, 106]. Acted emotions mean those generated by professional or non-professional actors who are required to give an utterance with certain emotion. Natural emotions refer to those generated in a natural conversation. Induced emotions are between natural and acted emotions, which are generated in a well-designed laboratory setting. These three categories of emotions demonstrate different characteristics [22]:

Acted emotion data

Acted emotion data have some obvious advantages [106, 9]. First, the cost for recording acted speech is relatively low. Second, emotion can be collected in a controlled

environment where noise can be reduced to minimum. Last, recordings can be annotated easily as emotions are pre-defined in the script. However, the disadvantages of acted recordings are also evident. One is that the quality of the acted emotion is greatly dependent on the actor. Second, the acted emotions may fail at eliciting all the physiological reactions that an authentic emotion would elicit.

Natural emotion data

Natural emotion data can come from TV talk-shows [34], call-centre recordings, lecture and meetings, children playing [3] and medical dialogue, etc. However, it is not easy to obtain high-quality and balanced natural dataset. First, specific emotions could be very rare under those situations. Second, the quality of recording may be low as there are many unexpected and uncontrollable factors in the environment. Third, emotions and their expressions may be shadowed by other normal activities in a natural setting. As a consequence, although natural emotion data has excellent qualities, it is challenging to obtain good-quality examples of them.

Induced emotion data

Induced emotion data are often obtained via the Wizard-of-Oz scenario, in which the human subjects are convinced they are interacting with a computer, while in fact, a human experimenter is operating the machine and returning answers according to the inputs of the user [4]. One characteristic of induced emotions is that they are able to blend some of the advantages of acted and natural emotions. The induced emotions are not only closer to reality as they are authentic, they are also easily annotated. However, some psychologists assume that the awareness of being inside a laboratory environment can affect the subjects displaying their emotions. In addition, it is hard to guarantee that emotions are actually elicited but not acted, and the elicited emotion is indeed the one that was hoped for. In spite of these drawbacks, induced emotions represent a valid and convenient alternative to natural emotions.

3.1.2 Emotion theories

Both categorical and continuous emotion theories may be adopted in creating emotion databases. Continuous emotions can have up to 3 dimensions, namely valence, arousal, and domination, respectively. Categorical emotions, on the other hand, can have many specific classes, and they are usually pre-defined. Vague emotional states may also be

specified to facilitate the annotation process, but in general only the common emotion classes are recommended for analysis.

3.1.3 Annotation scheme

Annotation is closely associated with the quality of emotion databases. Different database builders may use different annotation schemes, but they normally follow these conventions:

- Labels setting. A range of labels or classes are provided for use, but flexibility is also given for vague emotional states.
- Multiple annotators. There should be different experts in the field as annotators and their annotations will be estimated to ensure the consistency.
- Majority voting. Only the annotations agreed by a majority of the experts will be kept to maximise the reliability.

Taking the database IEMOCAP [10] as an example, the categorized emotion classes pre-defined are anger, happiness, sadness, neutral, disgust, fear, surprise, excite, frustration, and unknown. Ground truth labels are obtained by majority voting (i.e. at least agreed by two annotators). There are about 25.4% of the utterances labelled differently by three annotators. Because different classes of examples vary considerably in the size, in research, they may be further selected or combined.

3.1.4 Selecting emotion corpora

It seems ideal to acquire as many as possible databases for our research, but in reality, we have to consider the following factors regarding the selection of databases:

- Financial aspect. Commercial databases can be quite expensive and it is preferred to use those public free ones.
- License. It is important to obtain the license for use, and fortunately, there are some good databases available for academic purpose.
- Quality. Only those widely-accepted, i.e. frequently utilised in the research area, are chosen.

Table 3.1: The used databases/corpora

Database	Language	Speakers	Volume	Type	Emotions
EMODB	German	5M, 5F	800 sentences	acted	anger/fear/ joy/sadness/ boredom/ disgust/ neutral
SAVEE	English	4M	480 sentences	acted	anger/disgust/ fear/happiness/ sadness/ surprise/ neutral
Aibo	German	51 children	18216 sentences	natural	joyful/bored/ surprised/ motherese/ emphatic/ reprimanding/ angry/neutral/ touch/other
IEMOCAP	English	5M, 5F	7380 sentences	induced	anger/sadness/ happiness/ frustration/ excited/ neutral/ others

- Popularity. The popularity of the used data allows the comparison with other works to be easier.
- Size. Considering the time and computing resources available, data size should be also taken into account.
- Diversity. It is desirable to have diverse databases for convincing and general conclusions.

3.2 The used corpora

We have collected some well-known databases which are available to the public. A summary of them is seen in Table 3.1. All of these databases are free to public for the academic purpose, and they have been widely used in the area. The varying size of

these databases allows us to use them more flexibly in our research, and the requirement of diversity is also met as the databases involve completely different speakers in different experiment settings, in spite of the limited number of the used languages. More details about these databases are given below.

3.2.1 EMODB

This is an audio collection of emotional utterance developed between 1997 and 1999 by Burkhardt et al. at the Technical University Berlin [9]. It contains 7 basic emotions, namely anger, fear, joy, sadness, boredom, disgust, and the neutral state. A total of 10 non-professional actors' utterances were used. Each actor was required to utter 10 sentences and each sentence was repeated for every single emotion. Therefore the database contains 700 sentences (10 actors * 10 sentences * 7 emotions) plus 100 additional sentences as a backup.

3.2.2 Aibo

The Aibo database [98] contains recordings of children interacting with Sonys pet robot Aibo. It consists of induced speech data in German. The children were led to believe that Aibo was responding to their commands, whereas it was actually controlled by a human operator in a Wizard-of-Oz manner. The data were collected at two different schools, identified as Mont and Ohm, with 25 and 26 children speakers from each, respectively. Five expert humans listened to the speech data and annotated each word independently. Considering the large size of Aibo, in our work we sometimes treat the two parts, Ohm and Mont, as two separate datasets.

3.2.3 IEMOCAP

The IEMOCAP database [10] contains approximately 12 hours of audio-visual data recorded from five male and five female actors. The goal of the data collection was to elicit natural emotions within a controlled setting. This goal was achieved with two elicitation framework: scripts, and improvisation of hypothetical scenarios. These approaches allowed the actors to express spontaneous emotional behaviours driven by the context (as opposed to read speech displaying prototypical emotions). Several dyadic interactions of approximately five minutes were recorded, which were manually segmented into turns.

3.2.4 SAVEE

This is an audio-visual acted emotional database [48] annotated with Ekman's discrete theory of emotion with 7 emotions: anger, disgust, fear, happiness, sadness and surprise plus the neutral state. The database contains the recordings of 4 students (all male) belonging to the University of Surrey. In total, the database contains 480 recordings (4 students * 120 recordings).

Chapter 4

Domain Adaptation Related Work

This chapter focuses on the main problem we address, domain shift in speech emotion recognition, and the related literature work. Specifically, Section 4.1 gives the definition of domain shift and discusses its influence on speech emotion data. Section 4.2 and Section 4.3 introduce the traditional domain adaptation approaches and the novel approaches using neural networks, including deep learning methods featuring adversarial learning. Section 4.4 summarizes the so-far progress of domain adaptation for speech emotion recognition, and Section 4.5 discusses the limitations of existing domain adaptation approaches and the relationship with our work.

4.1 Domain shift problem

We first clarify the definition of domain shift and domain adaptation, as many similar and relevant concepts have developed from different research areas such as covariate shift, dataset shift, and transfer learning.

4.1.1 Definition

Humans can easily transfer the knowledge from one area to other less known areas. In machine learning, there are many occasions that the information (often in the form of data) regarding the target domain is insufficient, and it is thus desired to utilise the information from a different but related domain to help address the target task. Besides, most traditional machine learning methods work under the assumption that the training and test data should be drawn from the same feature space and the same distribution. When the distribution changes, it is often necessary to recollect the needed data and

re-build the model, e.g. typically in the applications of Web document classification [114] and sentiment classification [6].

Transfer learning [78] arises to address these issues. There are many terms and definitions relevant to transfer learning: learning to learn, life-long learning, knowledge transfer, inductive transfer, multitask learning, incremental learning, and self-taught learning [105, 12, 86, 78]. The famous survey paper [77] gives a widely accepted terminology and taxonomy of transfer learning, based on which we make the definition of domain shift.

Formally, a **domain** D consists of two components, the feature space $\mathscr{X}$ and the marginal probability distribution $P(X)$. A **task** T consists of two components, the label space $\mathscr{Y}$ and the objective predictive function $f(X)$ which can be written as $P(Y|X)$ from a probabilistic viewpoint. With a source domain D_s, the learning task on this domain can be denoted as T_s and the joint probability for the predictive function can be denoted as $P_s(X,Y)$. Correspondingly, for a target domain D_t, the learning task is denoted as T_t and the joint probability for the predictive function as $P_t(X,Y)$. Domain shift happens when $D_s \neq D_t$ or $T_s \neq T_t$. Assuming that the feature space and label space are same for the source and target domains, the definition of domain shift can be simplified as follows:

Definition 4.1: Domain shift happens when $P_s(X,Y) \neq P_t(X,Y)$ where s and t refer to source and target domain respectively.

Domain shift is also termed dataset shift [83], concept drift [20] or covariate shift [100] under certain circumstances (e.g. the marginal distribution discrepancy between different domains). Domain shift adaptation (or domain adaptation for short), as the name suggests, means compensating for the domain shift so the discrepancy between different domains can be reduced or eliminated. It should be pointed out that transfer learning is a more general concept describing all the technologies used to achieve knowledge transfer across domains, and domain adaptation can be regarded as a special case of transfer learning. A rigorous categorization on different cases of transfer learning is seen [78]. Here we give the definition of domain adaptation as follows:

Definition 4.2: Given a source domain D_s and learning task T_s, a target domain D_t and learning task T_t, **domain adaptation** helps improve learning the target predictive function in D_t using knowledge in D_s and T_s, when $T_s = T_t$, and D_t lacks sufficient information for learning. Specifically, if D_t contains only a few labelled data, it is termed **supervised domain adaptation**; or if D_t contains sufficient data but there is no label information, it is termed **unsupervised domain adaptation**.

4.1.2 Domain shift in speech emotion data

Domain shift exists in speech emotion data and leads to the challenging cross-corpora problems, i.e., a model trained with one corpus will degrade significantly when tested on a different corpus. There are many factors causing domain shift in emotion data. As shown in Figure 4.1, these factors include:

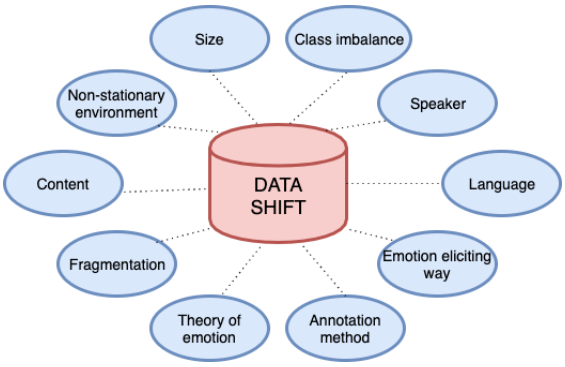


Figure 4.1: Factors causing data shift in speech emotion data

- Theory of emotions. The used theory determines whether the emotions are described by classes or dimensions. Generally categorical theories are more popular than continuous theories, but recently the latter are attracting growing attention because it provides a quantitative measure of the emotional states which cannot be simply classified as basic emotion classes [88, 87].

- Eliciting ways of emotions. The emotions used in different databases can be acted, natural, or induced. It has been shown that natural emotions are more difficult to recognize than others [106, 11, 111].

- Content. The utterances recorded can be pre-defined or directed deliberately towards certain topics, so the content of different corpora varies greatly.

- Language. The role of language in emotions is a large topic. According to [59], the emotions experienced by a given person depends on the emotion concepts available to that person, and the development of emotion concepts is closely related to the language. Therefore, some emotion concepts from one language may not have equivalents from other languages. For example, in Korea, han is the state of feeling sad and hopeful at the same time, and for Baining People of Papua New Guinea, awumbuk describes what you feel when your visitors leave [57].

- Subjects or speakers. As the corpora are acquired independently, the subjects could vary in the backgrounds, age, and professions. They may belong to different groups with certain characteristics, e.g., when actors are chosen as subjects, their performance could obviously differ from non-actors' performance.

- Size. Some corpora may contain relatively more recordings, and this poses the data imbalance issue (as well as class imbalance issue) in cross-corpora tasks.

- Fragmentation. The utterances are saved in different length in different corpora, and furthermore, various fragmentation skills can be applied to generate frames.

- Class imbalance. Human emotional states, are not evenly distributed in real life. In fact, neutral emotion is dominant in terms of frequency, making the useful emotionally coloured speech signals take up a very small portion in the whole. The degree of class imbalance vary across corpora due to different experimental designs.

- Annotation method. This is relevant to the adopted theory of emotions and to the specific annotation strategy. The annotators can be native or non-native, emotion-related experts or ordinary people, and their numbers can be different across emotion corpora.

- Recording conditions or non-stationary environment. These physical factors can have an influence even within one corpora and they are often uncontrollable.

All of these factors collectively contribute to the uniqueness of a database/corpus. In theory it is impossible to eliminate all these factors' influence to generate two perfectly matched corpora. Therefore it is unsurprising that domain shift exists extensively in speech emotion data. The consequence of domain shift can be directly observed through the degrading performance of a recognition model in a cross-corpora setting. It should be emphasized that although a corpus is often called a domain in our work, domain shift does not necessarily occur across different corpora. Within the same corpus, if there are different speakers or groups, domain shift also emerges across these speakers and groups. Simply speaking, if the dataset involves different speakers, domain shift is unavoidable.

4.2 Traditional domain adaptation approaches

4.2.1 Fine-tuning technique

Fine-tuning originally means taking weights of a trained neural network and using it as initialization for a new model being trained on data from the same domain (often e.g. images). It now usually refers to the process that takes a model that has already been trained for one given task and then tunes or tweaks the model to make it work on a second similar task. Therefore, fine-tuning can be regarded as a way of domain adaptation technique.

Fine-tuning allows us to take advantage of the knowledge learned by the trained model on new tasks. For example, the data size may be too small in the target domain, making it almost impossible to build a recognition system. Source domain can then be used to train a model which will be further tweaked or tuned by using the few target-domain examples so as to fit the target domain task better. In other words, fine-tuning can be achieved by making the existing model re-train/re-learn on the examples from the new domain.

4.2.2 Adaptive support vector machines

Adaptive support vector machines (SVM) [60] attempts to transform existing SVM classifiers into a new effective SVM classifier that would work on a new dataset with limited amount of labelled data. Though originally proposed to address image tasks, it also shows effectiveness on speech emotion recognition [1].

The approach works by minimising both the classification error over the training examples, and the discrepancy between the original and adapted classifier. The new optimization problem seeks a decision boundary close to that of the classifier trained from the source domain, while managing to separate the new labelled data from the target domain.

4.2.3 Importance weighting

This is a large category of domain adaptation solutions with the assumption that the conditional distributions are same but the marginal distributions are different between the source and target domains, i.e. $P_s(Y|X) = P_t(Y|X)$ and $P_s(X) \neq P_t(X)$. This special case of domain shift is usually termed covariate shift, and the typical approaches to covariate shift is importance weighting [100] which gives more weight to the examples

from the target domain. Covariate shift has been intensively studied in the literature and there have developed some useful techniques. Note that tackling covariate shift could be useful only on the premise that the support of test data are contained in the support of training data [100].

Formally, importance weight, denoted by β, is calculated by

$$\beta(x) = \frac{p_{te}(x)}{p_{tr}(x)} \tag{4.1}$$

where p_{tr} and p_{te} are the probability density of training samples from the source domain and test samples from the target domain, respectively. By introducing this ratio, the learning algorithm is pushed towards the more important regions in input space. As a result, the key to tackling covariate shift is reduced to the calculation of importance weights.

There are two approaches to calculating the importance weights. One is directly estimating the probability density functions of training and testing data. The other approach can determine the importance weight without attempting to estimate the densities. Examples of the first approach include histogram estimation and kernel density estimation. However, as the number of input dimensions increases, these direct estimation methods suffer from the curse of dimensionality and perform poorly especially when the available training data are limited. Therefore, the second approach is more practical. There are three solutions based on this idea which have been verified able to solve covariate shift [38]:

- Kernel means matching (KMM) [82]. It works by minimising the difference between the means of importance-weighted training and testing data distributions in a high-dimensional feature space. This new space is induced by a kernel function and there is no need for density estimation. The method allows to obtain importance estimation directly at the training input points. KMM is expected to work well even in the high-dimensional case.

- Unconstrained least-squares importance fitting [51]. It formulates the problem of finding importance weights as a least-square function-fitting problem.

- The Kullback-Leibler importance estimation procedure (KLIEP) [109]. It uses the divergence between the importance-weighted test distribution and the true test distribution in the terms of Kullback-Leibler (KL) divergence [8]. The biggest advantage of this method is that it relies on the testing not on training

data to estimate all optimization parameters. This is very useful in the scenarios when a large amount of testing data is available. There is only one parameter, namely the kernel width, which can be tuned by likelihood cross-validation.

All of the three solutions of importance weighting above are tested in [38] for speech emotion recognition under the setting of unsupervised domain adaptation, yielding good performance. However, it has been found that some novel neural networks based approaches [19, 18, 17], which will be discussed in the next section, can achieve better recognition accuracy than these addressing importance weighting.

4.3 Neural networks based adaptation approaches

Different to fine-tuning on neural networks which does not need to modify the architecture, some domain adaptation approaches based on neural networks are characterized by modifying the architecture, and particularly, by choosing a part of the architecture for information sharing between domains.

4.3.1 Models featuring autoencoders

The works by Deng [19, 18, 17] have tried using autoencoders, a special neural networks, for domain adaptation. A typical approach, adaptive denoising autoencoder (adaptive-DAE or A-DAE) [18] employs a more recent variant of autoencoders, denoising autoencoder (DAE). As viewed in Figure 4.2, in the basic architecture of DAE, an input example $x \in R^n$ is first converted to a corrupted version $\tilde{x}$ by adding some common noise, e.g. Gaussian noise or masking corruption (deleting random elements of the input). The hidden representation $h(\tilde{x})$ is

$$h(\tilde{x}) = f(W^{(1)} \cdot \tilde{x} + b^{(1)}) \tag{4.2}$$

where $f(\cdot)$ is a non-linear activation function, typically a logistic sigmoid function applied component-wise, $W^{(1)} \in R^{m \times n}$ is a weight matrix, and $b^{(1)} \in R^m$ is a bias vector. The network output maps the hidden representation h back to a reconstruction $y \in R^n$:

$$y = f(W^{(2)} \cdot h(\tilde{x}) + b^{(2)}) \tag{4.3}$$

where $W^{(2)} \in R^{n \times m}$ is a weight matrix, and $b^{(2)} \in R^n$ is a bias vector. Given a set of input examples χ, the DAE training corresponds to minimising the following objective

function:

$$J(\theta) = \frac{\lambda}{2}\left(\sum_{l=1}^{2}\sum_{j}\left\|w_j^{(l)}\right\|^2\right) + \sum_{x\in\chi}\|x-y\|^2 \tag{4.4}$$

where w_j^l is the j-th column vector of the l-th layer weight matrix $W_{(l)}$, and a weight-decay regularization term with hyper-parameter λ is included to avoid over-fitting. The minimisation is usually realized by stochastic gradient descent or more advanced optimization techniques such as L-BFGS [66] and conjugate gradient method [41].

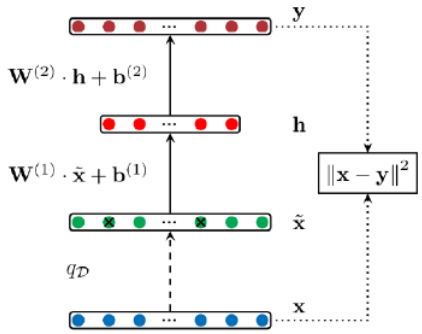


Figure 4.2: The architecture of denoising autoencoder (from [18])

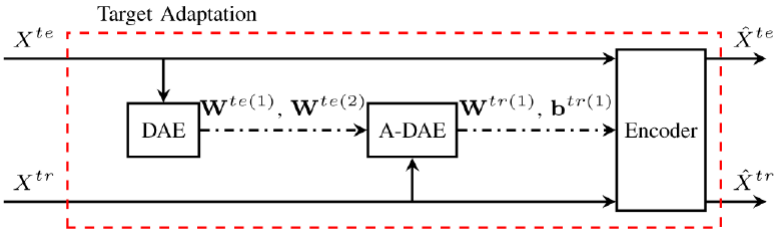


Figure 4.3: An example of adaptive autoencoders works [18] by forcing the weights adapt to the weights which were learned using unlabelled test data. The mismatch between the training and test data can be reduced in this way.

Figure 4.3 illustrates how adaptive-DAE (Figure 4.3) achieves domain adaptation. A DAE is first learned in a fully unsupervised way from the target domain data, resulting in the weight matrices ($W^{te(1)}$, $W^{te(1)}$) and bias vectors $b^{te(1)}$, $b^{te(1)}$). Then A-DAE forces their weights to adapt to these provided weights as well as minimising the reconstruction error between the input and output at the same time. Specifically, given a training example $x \in \chi^{tr}$, the objective function of an adaptive DAE is formulated as follows

$$J^{tr}(\theta) = \frac{\lambda}{2}\left(\sum_{l=1}^{2}\sum_{j}\left\|w_j^{tr(l)} - \beta w_j^{te(l)}\right\|^2\right) + \sum_{x\in\chi^{tr}}\left\|x-y^{tr}\right\|^2 \tag{4.5}$$

where the hyper-parameter β controls the transfer regularization. The weight matrices $W^{tr(1)}$ ($w_j^{tr(1)}$ is its j-th column vector) and $W^{tr(2)}$ are randomly initialized and learned during training, while the weights $W^{te(1)}$ and $W^{te(2)}$ are fixed during training. Using the weights $W^{tr(1)}$ and $b^{tr(1)}$ learned by the adaptive-DAE, the test data and training data can be encoded to form the representations suitable for standard supervised classifier (e.g. SVM) for speech emotion recognition.

We can observe that adaptive-DAE works by minimising the difference of weights between the training and target/test data. In fact, in the final stage, both the training and target data adopt the same encoder. The similar idea appears in other neural networks-based approaches [19, 17] which use shared-layers to realise domain adaptation.

4.3.2 Models featuring adversarial learning

Adversarial learning becomes popular after the introduction of general adversarial networks (GANs) [32] in 2014. In spite of the original motivation to synthesize pictures from random noise with a huge number of real pictures, the idea of adversarial learning turns out useful on domain adaptation as well, yielding better performance than traditional approaches. We first briefly review GANs and then discuss how adversarial learning is applied to domain adaptation.

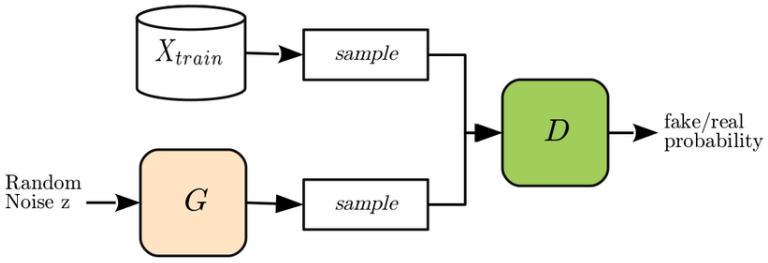


Figure 4.4: General adversarial networks

As shown in Figure 4.4, there are two key parts in GANs, the generator G and the discriminator D, which are engaged in a competing game. While G takes random noise as input, D takes both the source data and the generated representations from G as input. A discriminator tries to predict the domain label y_D (signifying whether the example is from source domain or from the generator) for any input sample, while a generator tries to fool the discriminator.

General adversarial networks suggests that the domain discrepancy can be implicitly reduced by adversarial learning. Therefore no measure for domain discrepancy is

needed. Typically, a work is seen in [110] which combines adversarial learning with discriminative representation learning for image classification.

Regarding supervised domain adaptation, because the target domain has only a few labelled data, some works also term this scenario as few-shot learning. The state-of-the-art few-shot learning approach, Few-shot Adversarial Domain Adaptation (FADA) [70] has two features: data pairs generation and binary adversarial discrimination.

With the source domain D_s and target domain D_t, data pairs generation is aimed at handling the scarcity of target data by pairing the target samples with source samples. In particular, four new groups of data pairs are created. The pairs of Group 1 consist of samples from the source distribution with the same class labels, while pairs of Group 2 have the same class label but come from different distributions (one from the source and one from the target distribution). Similarly, the pairs of Group 3 consist of samples from the source distribution with different class labels, and the pairs of Group 4 come from different class labels and different distributions (one from the source and one from the target distributions). The pairs from Group 1 and 2 are Positive pairs, while those from Group 3 and 4 are Negative pairs.

Binary adversarial discrimination is achieved by employing a discriminator (based on a neural network) that distinguishes the group labels of the data pairs. Specifically, the based network can be composed of a feature encoder G_e (parameterized by θ_e) and a prediction layer G_p (parameterized by θ_p). The network is first trained with source data D_s by minimising the following classification loss

$$L_c = -\sum_{x^s} y^s \log G_p(G_e(x^s, \theta_e), \theta_p) \tag{4.6}$$

where x^s is the example of the source domain and y^s is its corresponding label. Keeping G_e fixed, the discriminator G_D (parameterized by θ_D), with the Siamese architecture, is trained to discriminate the four groups by minimising

$$L_D = -\sum_{i} y_{g_i} \log G_D(g_i, \theta_D) \tag{4.7}$$

where g_i is one pair of examples with the group label y_{g_i}. The adversarial learning is realised by performing the following two steps alternately. The first step is updating

G_e and G_p by minimising

$$\begin{aligned} L_g = &-\lambda \sum \{ y_{g_1} \log G_D(g_2, \theta_D) + y_{g_3} \log G_D(g_4, \theta_D) \} \\ &- \sum_{x^s} y^s \log G_p(G_e(x^s, \theta_e), \theta_p) - \sum_{x^t} y^t \log G_p(G_e(x^t, \theta_e), \theta_p) \} \end{aligned} \tag{4.8}$$

where λ strikes the balance between classification and domain confusion. The second step is updating G_D by minimising Equation 4.7. Misclassifying Group 1 as 2, or misclassifying Group 4 as 3 means that the discriminator cannot distinguish Positive or Negative pairs between the source and target distributions. Therefore these two steps can satisfy the goals of domain confusion and class separability at the same time. The FADA approach is summarised in Algorithm 1.

Algorithm 1 FADA learning algorithm

1: Train G_e and G_p with D_s using Equation 4.6.
2: Uniformly sample data pairs with label y_{g_i} ($i \in \{1,2,3,4\}$) from D_s and D_t
3: Train G_D using Equation 4.7
4: **while** not convergent **do**
5: Update G_e and G_p by minimising Equation 4.8
6: Update G_D by minimising Equation 4.7
7: **end while**

FADA is the main comparative approach to our proposed CADA as it stands for the state-of-the-art supervised domain adaptation solution [70].

4.4 Progress and limitation

Some pioneer works have systematically evaluated cross-corpora speech emotion recognition with a number of high quality databases [94, 23]. These works agglomerate several corpora to form a source domain, but do not adopt more sophisticated adaptation techniques for reducing the domain shift. Yet they have shown that domain shift widely exist in speech emotion data, and combining as many as datasets in building source recognition systems can be a simple and effective method to address the cross-corpora problem.

Hassan [38] first treats the mismatch in emotional data as covariate shift and proposes compensating for that shift by classical importance-weighting at the instance level. At the feature level, some autoencoder-based transfer learning methods [17, 18] (which are briefly reviewed in Section 4.3.1) have developed to seek a shared feature

representation so that the knowledge can be transferred between the domains. All of these methods, however, are usually applied to unsupervised rather than supervised domain adaptation, which demands a lot of data in target domain that may not be easy to collect in reality.

With respect to supervised domain adaptation, [1] have revealed that even a few labelled data from the target domain can be hugely helpful. Specifically, [1] uses adaptive SVM as the adaptation scheme, which transforms exiting SVM classifiers into a new classifier by minimising both the classification error over the training examples, and the discrepancy between the original and adapted classifier.

It is noticed that although adversarial learning [27, 110] gains a great popularity on domain shift adaptation and achieve success on many challenging tasks, few works have applied adversarial learning to speech emotion recognition. As discussed above, a significant advantage about adversarial learning is that instead of directly measuring the similarity of different domains, it introduces a domain discriminator that distinguishes the source from the target domain, and a feature representation is then learned to be domain invariant by fooling the domain discriminator. So far, most of the adversarial learning based adaptation techniques have focused to address image-related tasks in the scenario of unsupervised domain adaptation. It interests us to attempt the use of adversarial learning on speech emotion recognition. In addition, we realize that supervised domain adaptation, a scenario often ignored, is highly meaningful for speech emotion recognition due to the difficulty of collecting a large amount of data. In fact, with few data from the target domain at hand, supervised domain adaptation can be a more practical solution than seeking more data from that domain towards unsupervised domain adaptation.

The state-of-the-art supervised domain adaptation approach FADA has demonstrated impressive performance in different applications [70]. Nevertheless, we find that it does not work well on speech emotion recognition. This is possibly because the pairing technique in the method cannot effectively deal with high intra-class variability, which is common in speech emotion data. In order to verify this hypothesis and solve the issue, we propose Class-wise Adversarial Domain Adaptation (CADA) and conduct toy dataset experiment to compare it with FADA in Chapter 5.

Chapter 5

The CADA Approach

In this chapter, a novel domain adaptation approach CADA featuring class-wise adversarial learning is proposed. Specifically, Section 5.1 depicts the supervised domain adaptation scenario, to which CADA is applied. Section 5.2 gives an intuitive explanation of CADA. Section 5.3 provides the technique details of the method, and Section 5.4 shows some illustrative exmaples of how CADA gains an advantage over the state-of-the-art approach FADA.

5.1 Supervised domain adaptation scenario

As collecting a large amount of labelled speech emotion data is difficult, how to make most use of the limited labelled data is a practical question. Unfortunately, these data alone are often not sufficient to build up a robust recognition system. One increasingly popular solution to this issue is domain adaptation, e.g., utilising the knowledge from a related and rich domain, i.e. source domain, to help improve the performance on the target domain.

Given the source domain D_s and target domain D_t (we use s and t to refer to the source and target domain, respectively), where the source domain follows the distribution $P_s(X,Y)$ and the target domain $P_t(X,Y)$ (X denotes the input speech and Y the emotional class), the goal of domain shift adaptation is to learn a classification function f that minimises the misclassification error $L_y(f(X_t),Y_t)$ by using all the data available in two domains. Under the setting of supervised domain adaptation (SDA), $D_s = \{(x_i^s, y_i^s)\}_{i=1}^N$ and $D_t = \{(x_i^t, y_i^t)\}_{i=1}^M$ ($M << N$). In other words, there are very limited labelled data from the target domain. Under the setting of unsupervised domain adaptation (UDA), $D_s = \{(x_i^s, y_i^s)\}_{i=1}^N$ and $D_t = \{(x_i^t)\}_{i=1}^M$, i.e., there are many

unlabelled target-domain data.

In speech emotion recognition, UDA is useful when annotating data can be difficult or expensive, while on the other hand, SDA is useful collecting data can be difficult or expensive. So far, SDA for speech emotion recognition has been rarely studied while most of the works have emphasized on UDA. Considering that annotating a few emotional speech examples can be more realistic than collecting a large quantity of examples, SDA is a very practical setting.

5.2 Intuition behind CADA

In exploring why few-shot adversarial domain adaptation (FADA) does not work well on speech emotion data, we find that this is possibly due to the high intra-class variability. As in FADA, adversarial learning is performed on data pairs generated from source and target domains. However, in the process of pairs generation, the specific emotion class information is not considered. As a result, the high intra-class variability, which is common in emotion datasets, is ignored and cause the adversarial learning between two domains to be less effective.

From the viewpoint of mathematics, domain shift adaptation methods usually work under the assumption $P(Y_s|X_s) = P(Y_t|X_t)$. By learning a feature space ϕ such that $P(\phi(X_s)) = P(\phi(X_t))$, it ideally leads to $P(Y_s|\phi(X_s)) = P(Y_t|\phi(X_t))$, which means the classifier can be shared by both domains. However, in speech emotion recognition, the underlying assumption that $P(Y_s|X_s) = P(Y_t|X_t)$ may be less solid because of the high intra-class variability between the source and target domains, and this further affects the learning process for the desired feature space.

To tackle this weakness, class-wise domain adaptation is aimed at seeking a feature space ϕ for $P(\phi(X_s)|y_i) = P(\phi(X_t)|y_i)$, $y_i \in Y$ instead of $P(\phi(X_s)) = P(\phi(X_t))$. With the assumption that $P(Y_s) = P(Y_t)$, by Bayesian theory, we wish to have

$$\begin{aligned} P(y_i|\phi(X_s)) &= \frac{P(\phi(X_s)|y_i)P(y_i)}{\Sigma_i P(\phi(X_s)|y_i)P(y_i)} \\ &\approx \frac{P(\phi(X_t)|y_i)P(y_i)}{\Sigma_i P(\phi(X_t)|y_i)P(y_i)} \\ &= P(y_i|\phi(X_t)) \end{aligned} \tag{5.1}$$

where $\approx$ is carried out for domain shift adaptation.

5.3 Technical details

The main principles in designing new domain adaptation scheme is to achieve class-wise adaptation via adversarial learning. Adversarial learning, as we have known from Chapter 4, is usually realised by introducing a domain discriminator, and the adaptation should be achieved by using a feature encoder/learner, similar to the shared-layers of the modified autoencoders [17] in terms of function that transfers knowledge across domains. It is also noticed that the feature learner is usually separated from the prediction/output part. In other words, the three components for domain adaptation methods are feature encoder, label predictor, and discriminator. The combination of label predictor and feature encoder can be regarded as a classifier. Furthermore, the feature encoder should be shared by the predictor and the discriminator, as shown in many adversarial learning based domain adaptation approaches.

In spite of the normal practice of domain adaptation featuring adversarial learning which constructs a classifier and a discriminator separately (with feature encoder shared) [27, 110], it seems unnecessary to have separate classifier and discriminator for adversarial learning. In fact, it could be cumbersome to learn different discriminators for each class for class-wise adaptation. A different but intuitive thought is directly combining the class discrimination and domain discrimination into one process. To that end, we try to build a domain-class discriminator (DCD) which can not only distinguish the classes but also distinguish the domains. Correspondingly, the output/predictions of this DCD should contain both the class information and the domain label information, and thus we need new labels for all the data.

Taking a binary classification task as an example, all the original data can be re-categorized into four new groups that fit the training of DCD. The new labels of these groups are: d_1 indicating Class 1 from source domain, d_2 Class 2 from source domain, d_3 Class 1 from target domain, and d_4 Class 2 from target domain. In the testing stage, we perform classification with these 4 categories and treat either the prediction d_1 or d_3 as Class 1, and either d_2 or d_4 as Class 2. This categorization scheme can be straightforwardly popularized to the cases with more classes.

Now we can give the model structure of the new class-wise adversarial domain adaptation approach as shown in Figure 5.1. It comprises a feature encoder and a predictor. The basic architecture of DCD can be a multi-layer perceptron (MLP) by simply modifying its output to adapt to the newly-assigned group labels. The hidden layer of the MLP is thus trained to learn both discriminative and domain-invariant features.

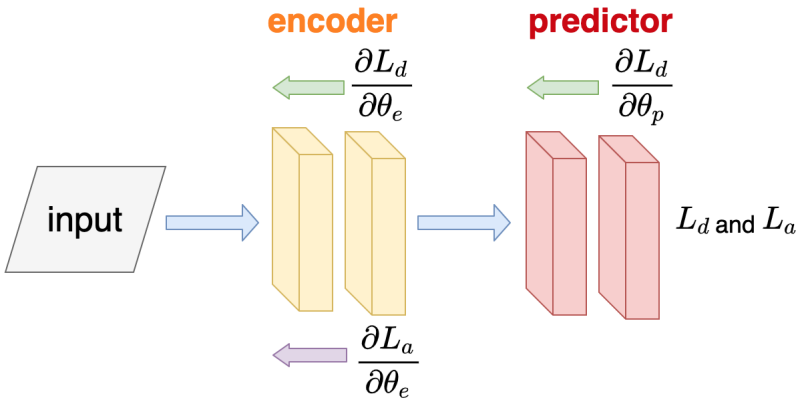


Figure 5.1: The class-wise adversarial learning domain adaptation (CADA) structure. It comprises a feature encoder and a predictor, parameterized by θ_e and θ_p respectively. The training process consists of two stages. In the first stage, both the encoder and predictor are trained based on the loss function Ld defined in Equation 5.2. In the next stage, the predictor is fixed and only the encoder is trained based on the loss function defined in Equation 5.4.

As shown in Figure 5.1, CADA comprises two components, the feature encoder G_e (parameterized by θ_e) and the predictor G_p (parameterized by θ_p). The model is first trained to distinguish the group labels. In case of binary class, 4 new groups are formed. To achieve that, both θ_e and θ_p are updated to minimise the typical cross entropy loss function

$$L_d = -\sum_{i}^{N+M} d^{x_i} \log G_p(G_e(x_i, \theta_e), \theta_p) \tag{5.2}$$

where d^{x_i} is the category of x_i. This step equips the model with the basic discriminative ability for the four groups. Meanwhile, to achieve domain adaptation, the features represented by the hidden-layer in the model should learn to be domain-invariant. To that end, θ_e is also updated by minimising the following loss function

$$\begin{aligned} L_a = -\{ & \sum_{x \in X_{d_1}} d_3 \log G_p(G_e(x, \theta_e), \theta_p) \\ & + \sum_{x \in X_{d_2}} d_4 \log G_p(G_e(x, \theta_e), \theta_p) \\ & + \sum_{x \in X_{d_3}} d_1 \log G_p(G_e(x, \theta_e), \theta_p) \\ & + \sum_{x \in X_{d_4}} d_2 \log G_p(G_e(x, \theta_e), \theta_p) \} \end{aligned} \tag{5.3}$$

where X_{d_i} denotes all the examples belonging to d_i ($i \in \{1,2,3,4\}$). This step is designed to encourage the confusion of the equivalent classes in different domains. Specifically, we want the model to believe the examples of certain class in one domain also belong to the equivalent class in the other domain. For instance, the first term on the right side of Equation 5.3 suggests that the examples from d_1 are also from d_3. This principle is applied to all categories we defined. The designed two steps perform alternately until some pre-set conditions are met, e.g., the maximum number of training epochs is reached.

While FADA [70] performs adversarial learning on newly-generated data pairs without considering specific class information, minimising Equation 5.3 allows the adversarial learning to operate on each specific common class across the domains, i.e., realising class-wise adversarial learning. Ideally, the model trained in this way can distinguish the class information but not the domain information, and thus, is applicable to the target domain.

Algorithm 2 CADA learning algorithm

1: Initialize θ_e and θ_p randomly
2: Re-label training examples of k classes in both source and target domains in terms of d_i, $i \in \{1,2,...,k,k+1,...k+k\}$
3: **while** not convergent **do**
4: Update θ_e and θ_p by minimising Equation 5.2.
5: Update θ_e by minimising Equation(5.4).
6: **end while**

It is rather straightforward to apply CADA to a multi-class case by modifying the output layer as shown in Figure 5.2. The loss function guiding the adversarial training accordingly changes to Equation 5.4 where k refers to the class number. The new category label $d_i(1 <= i <= k)$ corresponds to the i-th class in the source domain and $d_{k+i}(1 <= i <= k)$ corresponds to the i-th class in the target domain. For clarity, the CADA learning process is summarized in Algorithm 2.

$$
\begin{aligned}
L_a = -\{ & \sum_{x \in X_{d_1}} d_{k+1} \log G_p(G_e(x, \theta_e), \theta_p) \\
& + \sum_{x \in X_{d_2}} d_{k+2} \log G_p(G_e(x, \theta_e), \theta_p) + \ldots \\
& + \sum_{x \in X_{d_k}} d_{k+k} \log G_p(G_e(x, \theta_e), \theta_p) \\
& + \sum_{x \in X_{d_{k+1}}} d_1 \log G_p(G_e(x, \theta_e), \theta_p) \\
& + \sum_{x \in X_{d_{k+2}}} d_2 \log G_p(G_e(x, \theta_e), \theta_p) + \ldots \\
& + \sum_{x \in X_{d_{k+k}}} d_k \log G_p(G_e(x, \theta_e), \theta_p) \}
\end{aligned}
\tag{5.4}
$$

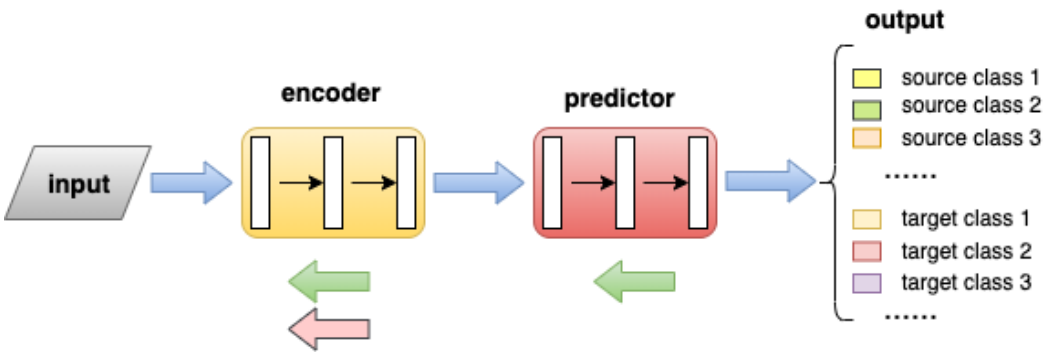


Figure 5.2: CADA in the multi-class case. It is characterized by a modified output layer. Accordingly the adversarial training operates for all common classes between the source and target domains.

Although we illustrate CADA structure with an MLP as the basis, it is worth pointing out that deep learning models can also serve as the basic of CADA. Rather straightforwardly, in the feature encode, hidden layers in MLPs can be replaced with convolutional layers and pooling layers, with the fully-connected layer still as the output layer. Then CNN-based CADA can be trained following similar procedures as given in Algorithm 1.

In the situation of unsupervised domain adaptation when no labelled target data are available, we wonder whether it is possible to utilise the pseudo-labels for class-wise domain adaptation. Then the question is how to generate the proper pseudo-labels. One natural solution is that we may exploit the source model to predict the unlabelled

examples in the target domain. However, considering the data shift, the predictions on the target data by the source model seem unreliable. In fact, in the extreme case, all the target data can be treated as belonging to the same class by the source model. Then CADA cannot work as there are only one class in the target domain. Alternatively, we may utilise the prediction confidence to seek the suitable pseudo-labelled examples.

Given a N-class problem, the prediction with confidence of one testing example can be written as a vector $[c_1, c_2, ..., c_N]$ where $\sum_i c_i = 1$. Because CADA needs the examples of all classes, one strategy to identify the k examples belonging to Class j is to sort $\{c_j^1, c_j^2, ..., c_j^M\}$ in a descending order, where M stands for the number of all predicted examples, and to select the top k values which correspond to the wanted examples. Based on this idea the algorithm of CADA for unsupervised domain adaptation (denoted as u-CADA) is given below.

Algorithm 3 u-CADA learning algorithm

1: Train the source model H with all source labelled data X^s
2: Predict the target examples X^t with H and get the confidence matrix M
3: Select k examples with the highest confidence value in each class and get the pseudo-labelled data $\hat{X}^t$
4: Use X^s and $\hat{X}^t$ for CADA

5.4 Illustrative examples

In cross-corpora problems, it is often found that a model trained on the source corpus predicts most of the target examples as the same class. In order to simulate such a scenario, we design a simple toy dataset as shown in Figure 5.3, where almost all the target examples ('+') are classified as the same class (blue) by the source model. We further choose a random target example from each class to represent the known target examples (shown in Figure 5.4), which are used to adapt the source model. Based on this design, we can compare the performances of different domain adaptation techniques.

In addition to FADA and CADA, we consider two adaptation schemes for comparison: 1) fine-tuning, which directly uses the target data available to tune the source model as discussed before, and 2) mix-tuning, which mixes all the source and target data available to perform regular supervised learning. We draw the changed decision boundary after domain adaptation in Figure 5.4. Note that mix-tuning generates a decision boundary exactly as in Figure 5.3. This is understandable as the known target

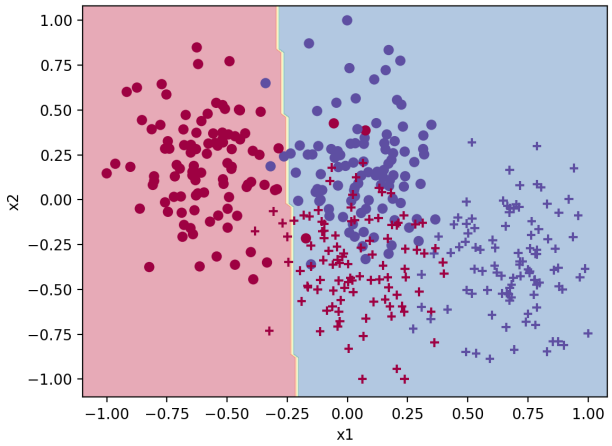
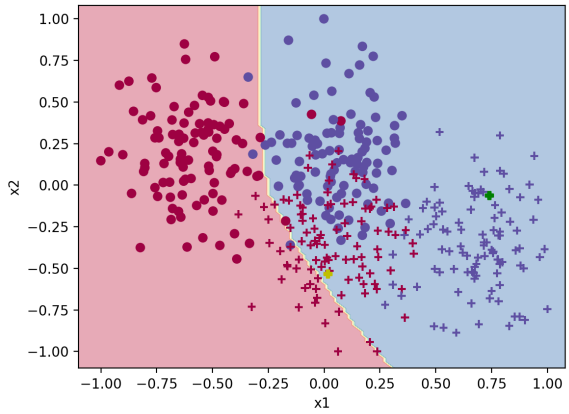


Figure 5.3: Toy dataset where source domain examples are represented by 'o' and target examples by '+'. The classes are distinguished by red and blue. As shown by the decision boundary of the model trained on source domain, nearly all target domain examples are classified as blue.

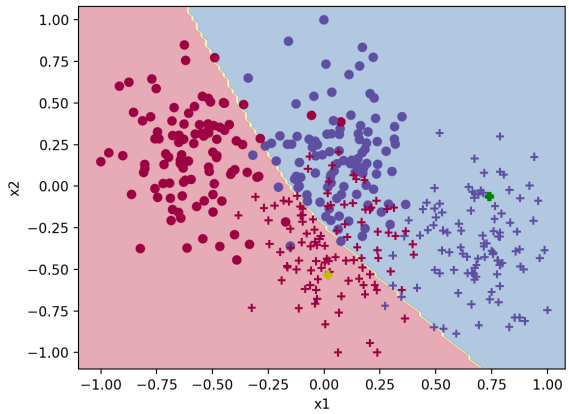
data is much less than the source data, and consequently, the minority is ignored considerably in training. On the other hand, as shown in Figure 5.4c, fine-tuning achieves the best separation between the two known target examples (yellow and green points) while the performance on the source data is sacrificed. Regarding FADA, we can find that from Figure 5.4a, it maintains a good classification on the source domain but it does not classify the known target examples correctly. Last, we can find that CADA is the only method that keeps good performance on both the source domain and the known target data. Clearly, fine-tuning is a pro-target method and mix-tune a pro-source method. While FADA is more similar to pro-source, CADA achieves a better balance between pro-target and pro-source. It seems that such a property of FADA (i.e. being more pro-source) makes it less effective when high intra-class variability occurs. Figure 5.5 gives another example on how FADA fails to capture the domain shift and how CADA successfully adapts to the target domain. It should be noted that, however, the toy datasets are only 2-dimensional. For data in high-dimensional in space, the intra-class variability can be very complex, and FADA may have better performance when the source and target domains share the high-level feature representations.

5.5 Summary

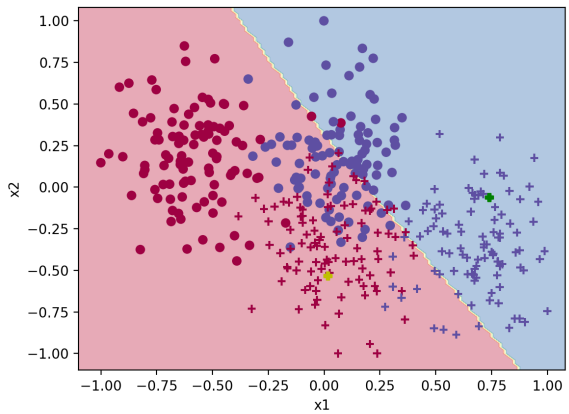
This chapter focuses on the approach termed Class-wise Adversarial Domain Adaptation, the main contribution of this thesis. The approach is applicable to supervised domain adaptation, and its advantage over the state-of-the-art techniques is due to the fact that it explicitly eliminates the domain shift for all common classes while the other approach cannot deal with the high intra-class variability in speech emotion data. We verify this hypothesis with toy dataset illustrations. Another advantage of the proposed approach is its straightforwardness and simplicity thanks to the compact structure which can perform classification and domain confusion at the same time.



(a) FADA

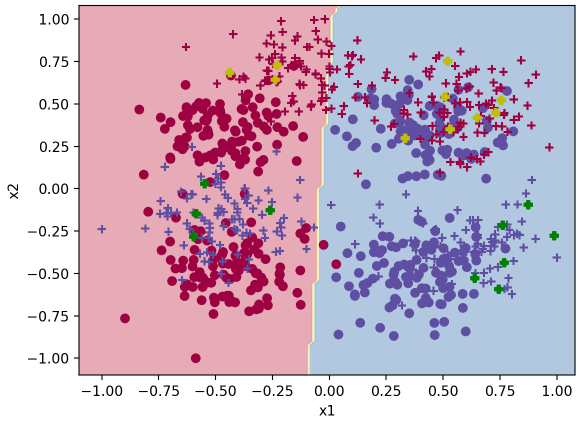


(b) CADA

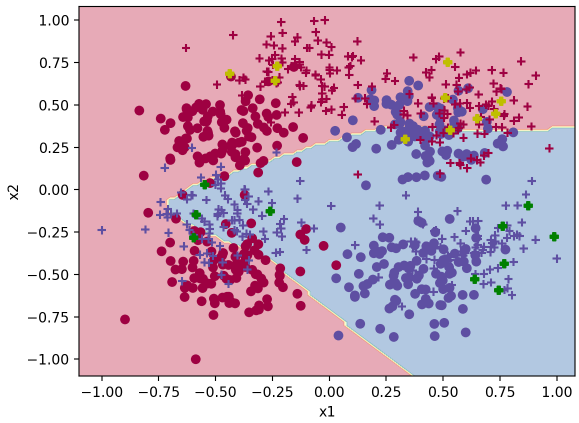


(c) Fine-tuning

Figure 5.4: Decision boundaries after domain adaptation by FADA, CADA, and fine-tuning. The yellow and green points denote the known target examples from the red and blue classes, respectively. Notice that all other target data (blue and red '+') were not present in training or adapting the source model. Only CADA keeps good performance on both source and known target data, suggesting it achieves a balance between the pro-source method FADA and the pro-target method fine-tuning.



(a) FADA



(b) CADA

Figure 5.5: Decision boundaries after adaptation using the known target examples (yellow and green points). Although each domain is simply decomposed of four Gaussian clusters, the intra-class distributions vary considerably between the two domains. As shown above, FADA cannot capture the domain shift and therefore fails to utilise the target examples for adaptation.

Chapter 6

MLPs-based CADA Evaluations

In this chapter we give a systematic evaluation on the proposed CADA approach. We first introduce the basic experiment design in Section 6.1, then we present various experiments including cross-corpora experiment in Section 6.2, intra-corpus experiment in Section 6.3, and the unsupervised domain adaptation experiment in Section 6.4.

6.1 Experiment design

6.1.1 Task-setting principles

In order to have a comprehensive evaluation on the proposed method, we have formed different kinds of tasks considering the variety of emotion corpora. (In experiment stage, we also use the terms dataset/datasets besides corpus/corpora). The sizes and emotion classes of these corpora are shown in Table 6.1. Specifically, the following factors are in consideration for experiment design.

- Cross-corpora or intra-corpus setting. Cross-corpora problems involve different corpora that demonstrate great domain shift between them. Similarly, domain shift also naturally occurs within one corpus because of the different subjects or other conditions. Based on the characteristics of the datasets summarized in Table 6.1, for cross-corpora setting, we adopt the larger dataset Aibo or IEMOCAP as the source domain, and the smaller-sized dataset EMODB or SAVEE as the target domain. It is consistent with our daily experiences that the source domain needs to be sufficiently informative to help address the target domain of which only little knowledge is known. For intra-corpus setting, IEMOCAP is the ideal

dataset because it consists of five similar-sized separate sessions involving different subjects. By splitting these sessions into the source and target domain, we can evaluate how domain shift occurs and how domain adaptation approaches work within one corpus.

- Basic or general emotion classes. We refer to prototype emotions as basic emotions such as anger, sadness, and happiness. These basic emotions are often used to label the examples (generate the classes) in each dataset. As shown in Table 6.1, the contained basic emotion classes are highly different across the corpora in terms of category and size. Therefore it is more reliable to select the common emotion classes from different datasets to form the source and target domains. In addition, we adopt general emotion categories Positive and Negative to encompass all of the basic emotion classes. On one hand, it allows us to exploit all the data contained in each dataset. On the other hand, it provides another meaningful classification of emotions as we may not care about the precise emotional state, and moreover, in practice some emotions can be too complex to be described by one of those prototype emotions.

- Binary-class or multi-class tasks. It is important to evaluate if the proposed approach suits multiple-class tasks, or more precisely, if the performance of the approach can be heavily influenced by the task complexity with respect to class number.

- Simple or deep neural networks. To observe the scalability of the approach, MLPs and CNNs are the typical examples of simple and deep neural networks respectively, so our approaches will be built based on MLPs and CNNs.

- Speaker dependent or independent setting. Both settings are tested to generate a comprehensive evaluation, and they may also give us insights in understanding the applicable conditions of the domain adaptation approaches.

- Emotion class balance. In real life emotion classes distribution can be highly imbalanced, and such a phenomenon is also reflected in the datasets. In training models with source domain, to avoid the influence of class imbalance, we may use a sample of the data of certain classes, assuming the data is sufficient. When the data is not sufficient, we try to avoid the cases where the size ratio of different classes is over 2.

Table 6.1: Emotion classes and sizes

	EMODB	Mont-Ohm	IEMOCAP	SAVEE
anger	127	611-881	1103	60
sadness	62	x	1084	60
happiness /excitement /joy	71	215-674	595 /1041	60
surprise	x	x	107	60
fear/anxiety	69	x	40	60
disgust	46	x	2	60
boredom	81	x	x	x
frustration	x	x	1849	x
neutral	79	5377-5590	1708	120

- Model's performance on source domain. The model trained on source domain should have a proper performance prior to being adapted for the target domain. Intuitively, it is almost impossible for a unfit source-domain model to perform well on a different domain (unless extremely lucky).

- Difficulty of the task. It is wise to start with easier tasks and then move on to more difficult ones. For example, using datasets with smaller size or fewer classes at the beginning are usually helpful for experiment design.

Based on the considerations, the speech emotion recognition tasks designed in our experiments are divided into the following parts.

- Cross-corpora basic-emotion binary-class tasks.

- Cross-corpora basic-emotion multi-class tasks.

- Cross-corpora general-emotion tasks.

- Intra-corpus speaker-dependent setting.

- Intra-corpus speaker-independent setting

All of them will be tested with MLPs-based and CNNs-based CADA, and this Chapter covers the experiments with MLPs-based CADA. Note that all the experiments are conducted on TensorFlow in python.

6.1.2 Features and models

We consider traditional hand-crafted features for building simple MLPs models. Particularly the GeMAPS feature set (62-dimensional) is chosen for the good balance between feature size and performance it achieves. More details of the GeMAPs can be found in Chapter 2. With GeMAPS, MLPs are used as basis for domain adaptation methods.

6.1.3 Baselines and comparative approaches

We use regular supervised learning methods (without domain adaptation) to establish baselines as follows. These baselines provide us a reference of accuracy that can be later compared with the accuracy by domain adaptation methods.

- *all-source*: using the trained source model without any target information (no any adaptation) for prediction on target domain;
- *label-target*: using only the labelled target data (no source domain knowledge used) to train and test. This baseline requires the labelled data in the target domain be sufficient to establish a model, thus it is only used under certain conditions.

All of the methods above are implemented via MLPs. Besides these baselines, the domain adaptation methods for comparison with CADA are

- Fine-tuning (or FT), which means building a model with source data and further tuning the model with the target data available. This is a famous trick in training neural networks and can be treated as the most straightforward domain adaptation technique.
- FADA. The state-of-the-art domain adaptation approach featuring adversarial learning. Technical details of FADA has been presented in Chapter 4.

6.1.4 Model selection

For domain adaptation, due to the lack of target-data information, model selection is performed on the source domain based on 5-fold cross validation for all the methods except the baseline *label-target*, for which model selection is based on the target domain. Then the selected model hyperparameters are used by the domain adaptation

source	target	emotion class
Ohm	EMODB	anger, happiness
Ohm	SAVEE	anger, happiness
Aibo	EMODB	anger, happiness
Aibo	SAVEE	anger, happiness
IEMOCAP	EMODB	anger, happiness
IEMOCAP	EMODB	anger, sadness
IEMOCAP	EMODB	sadness, happiness
IEMOCAP	SAVEE	anger, happiness
IEMOCAP	SAVEE	anger, sadness
IEMOCAP	SAVEE	sadness, happiness

Table 6.2: Cross-corpora binary basic-emotion tasks

methods. For instance, if the number of the hidden-layer neurons in the source model MLP is set N, we require that the feature-encoder layer in CADA has N neurons, and in FADA, the hidden-layer of the basic source-domain model has N neurons, and the hidden-layer in the discriminator has $2N$ neurons because it takes the pairs of the source-domain features as input. This practice can ensure the performances of different methods are decided by the learning algorithm instead of the model complexity as they adopt the similar model architecture.

6.2 Cross-corpora experiment

6.2.1 Basic-emotion binary-class tasks

Setting tasks

The larger datasets, Aibo (either the part Ohm or Mont), and IEMOCAP are used as source domain, and the smaller datasets, EMODB and SAVEE, as target domains. Furthermore, considering that 1) the target and source domains should have the same emotion classes, and 2) the emotion classes are desirably balanced in size (to minimise the influence of class imbalance), we set the specific tasks as summarized in Table 6.2. (For brevity of the tables later, we may simplify anger as ang, happiness as hap, etc.)

Model selection

For preprocessing and feature extraction, we use GeMAPS (62 features) [24] extracted by OpenSMILE and normalize all the features by mapping the values to the range

Hyper-parameter	Range
hidden-layer	$\{1, 2\}$
hidden-layer neurons	$\{32, 64, 128, 256, 512\}$
batch-size	$\{16, 32, 64\}$
epochs	$\{100, 200, 300, 400, 500\}$

Table 6.3: Model selection for cross-corpora binary basic-emotion classes tasks

$[-1, 1]$. The range of hyper-parameters for training the source-domains are shown in Table 6.3. Besides, all models use the Adam as optimizer with default learning rate, and use relu as the activation function for the hidden-layers and softmax as the activation function for the output-layer. Loss function is categorical cross-entropy.

The choices of hyper-parameters of the source-domain models as well as the corresponding accuracy is reported in Table 6.4. All models are 1-hidden-layer MLPs. Standard deviation generated from 5-fold cross validation is listed with the accuracy. From Table 6.4, it is observed that

1. Even for very basic 2 emotion classes, the recognition performance of machine learning (MLPs) models can be much lower than our (human) expectation, e.g. for anger and happiness in the data IEMOCAP, only 59.1% is achieved (a random guess is 50%);

2. Results using different datasets vary considerably. Aibo (including Ohm) are relatively easier to process than IEMOCAP for recognition between anger and happiness. Different binary class tasks within the same dataset can be highly different in terms of difficulty (reflected in the performance);

3. The standard deviation of accuracy from cross-validation is relatively large. This suggests the disturbance of the used data in training the model has a great influence on the model's generalization performance. In other words, there is an obvious domain shift in that dataset.

Result and analysis

For domain adaptation, the used target-domain examples are randomly selected with the same setting. That means all the domain adaptation methods use the same examples in each trial (an essential practice for fairness). Particularly, we set the number of the used target data from 2 and increase the number gradually (specific values seen in

Table 6.4: Hyper-parameter choice and accuracy of source models for the cross-corpora binary basic-emotion classes tasks

Domain	Emotions	Hyper-parameters	Accuracy %
Ohm	ang hap	neurons 256; batch 32; epochs 100	79.5 ± 5.8
Aibo	ang hap	neurons 128; batch 32; epochs 300	77.4 ± 10.5
IEMOCAP	ang hap	neurons 64; batch 32; epochs 200	59.1 ± 18.7
IEMOCAP	ang sad	neurons 64; batch 32; epochs 200	85.7 ± 4.5
IEMOCAP	sad hap	neurons 64; batch 32; epochs 200	68.3 ± 16.4

the first row of the tables reporting the results). 20 trials will be conducted for each adaptation method in our experiment. Unweighted accuracy (UA) is reported, as it can reflect the overall performance equally on all classes and is recommended in many literature works.

The experiment results are presented in Table 6.5 and Table 6.6. In the tables, the first column lists the task information including the used source-domain dataset, the emotion classes and the baseline source. In the second column, FT stands for fine-tuning, one of the three adaptation schemes. The digits in the first row indicates the number per class of the target-domain examples used for domain adaptation. For all the accuracy values, the standard deviation is relatively large (between 3.1 and 7.2). Because the standard deviation values are similar for the comparative methods, the specific values are not shown in the tables. From Table 6.5-6.6, we can make the following points.

- All of domain adaptation approaches achieve better performance than the no-adaptation baseline (*all-source*) when only a few target examples are available. With the used target examples increasing, the accuracy of the adaptation methods becomes higher and then gets stable after reaching some point. This indicates the source model has been 'fully' adapted by target examples, or in other words, adding more target examples cannot provide more information about the target-domain class boundary.

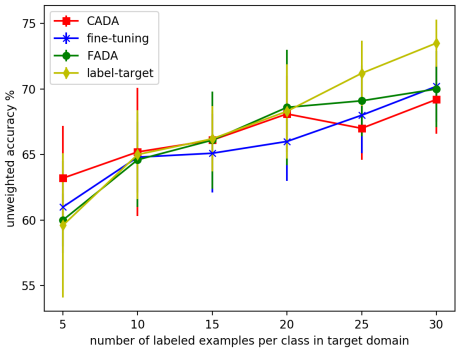
- Among the domain adaptation approaches, CADA clearly outperform the comparative approaches as in most cases (highlighted in the tables), it is the winner in performance. Fine-tuning also demonstrates an advantage to FADA, which in some cases, only beats the baseline by a small gap. As we have discussed in Chapter 5, this is due to the high intra-class variability in emotion data which cannot be alleviated by the adversarial learning in FADA.

Table 6.5: Unweighted accuracy (%) when using EMODB as target domain in **cross-corpora basic-emotion binary-class** experiment. The numbers in the head row represent the amount of used target-domain examples per class for adaptation. P-value of t-test is also provided to ensure the difference of the top two larger means of accuracy by the three methods is on a significant level.

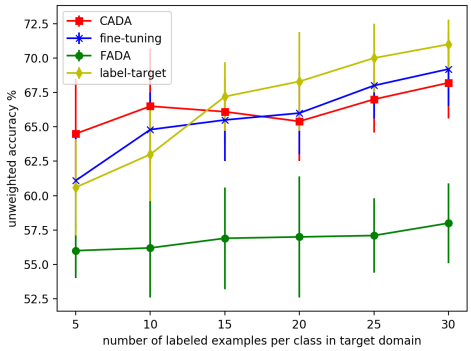
source	scheme	2	4	6	8	10	15	20
Ohm	FT	54.6	60.7	62.6	63.4	64.8	65.1	66.0
(ang, hap)	FADA	57.0	59.9	61.9	62.4	64.6	66.1	68.6
all-source: 54.2	CADA	**58.8**	**62.4**	**65.6**	**66.3**	**65.2**	**66.1**	**68.1**
t-test	p-value	0.02	0.04	0.02	0.01	0.01	0.08	0.05
Aibo	FT	58.1	60.2	63.1	63.7	64.8	65.5	**66.0**
(ang, hap)	FADA	55.2	55.9	56.1	56.2	55.9	56.9	57.0
all-source: 54.8	CADA	**60.0**	**64.0**	**65.4**	**66.2**	**66.5**	**66.1**	65.4
t-test	p-value	0.02	0.04	0.02	0.02	0.07	0.03	0.08
IEMOCAP	FT	60.3	63.7	64.2	64.8	**67.2**	69.0	**68.5**
(ang, hap)	FADA	58.9	59.0	60.0	60.3	61.1	62.3	62.4
all-source: 58.1	CADA	**63.6**	**66.2**	**67.8**	**69.0**	67.1	**69.4**	67.5
t-test	p-value	0.03	0.04	0.02	0.02	0.01	0.04	0.06
IEMOCAP	FT	96.7	98.2	98.3	98.1	98.6	99.0	**99.1**
(ang, sad)	FADA	95.9	96.1	97.2	97.9	98.0	98.1	98.2
all-source: 95.2	CADA	**98.4**	**98.4**	**98.5**	**98.8**	**99.0**	**99.1**	98.9
t-test	p-value	0.02	0.01	0.03	0.07	0.04	0.05	0.06
IEMOCAP	FT	89.1	92.4	93.3	94.8	**94.0**	**94.1**	94.2
(sad, hap)	FADA	89.0	90.7	91.1	91.2	91.4	92.0	92.1
all-source: 87.6	CADA	**91.1**	**94.3**	**94.4**	**95.6**	93.8	93.7	**94.8**
t-test	p-value	0.11	0.05	0.03	0.07	0.01	0.03	0.04

Table 6.6: Unweighted accuracy (%) when using SAVEE as target domain in **cross-corpora basic-emotion binary-class** experiment. The numbers in the head row represent the amount of used target-domain examples per class for adaptation. P-value of t-test is also provided to ensure the difference of the top two larger means of accuracy by the three methods is on a significant level.

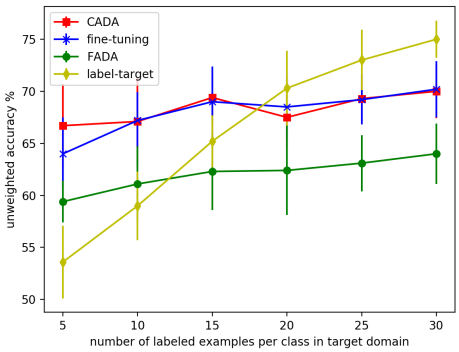
source	scheme	2	4	6	8	10	15	20
Ohm	FT	59.1	62.3	64.7	65.6	66.1	66.3	66.2
(ang, hap)	FADA	58.9	59.0	60.1	60.3	61.2	61.4	62.1
all-source: 58.3	CADA	**60.9**	**64.6**	**66.4**	**67.0**	**66.8**	**66.6**	**66.4**
t-test	p-value	0.04	0.06	0.09	0.02	0.01	0.01	0.03
Aibo	FT	58.9	61.3	63.2	64.7	65.5	67.9	**68.3**
(ang, hap)	FADA	56.9	57.3	57.8	58.7	59.2	59.9	60.3
all-source: 56.7	CADA	**60.2**	**63.8**	**65.5**	**66.4**	**65.5**	**68.7**	67.9
t-test	p-value	0.06	0.03	0.04	0.07	0.02	0.05	0.05
IEMOCAP	FT	57.9	61.2	64.4	**64.3**	65.5	67.9	68.0
(ang, hap)	FADA	58.0	58.3	58.4	59.1	59.4	60.1	61.5
all-source: 57.6	CADA	**59.8**	**62.9**	**66.8**	63.5	**66.1**	**68.9**	**68.1**
t-test	p-value	0.04	0.04	0.06	0.12	0.01	0.05	0.08
IEMOCAP	FT	73.2	77.8	78.9	79.2	80.0	80.9	81.1
(ang, sad)	FADA	72.2	72.8	73.9	74.2	73.4	73.9	74.4
all-source: 72.1	CADA	**75.9**	**79.2**	**79.5**	**80.2**	**80.0**	**81.1**	**81.3**
t-test	p-value	0.03	0.04	0.02	0.02	0.01	0.04	0.06
IEMOCAP	FT	75.3	77.8	79.0	80.3	81.2	83.4	**84.1**
(sad, hap)	FADA	75.2	76.0	76.5	77.1	77.2	78.9	79.9
all-source: 75.0	CADA	**76.3**	**79.0**	**81.9**	**82.4**	**83.7**	**84.2**	83.2
t-test	p-value	0.07	0.03	0.06	0.02	0.05	0.09	0.12



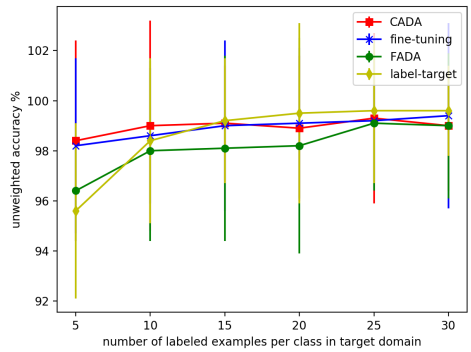
(a) Ohm (ang hap)



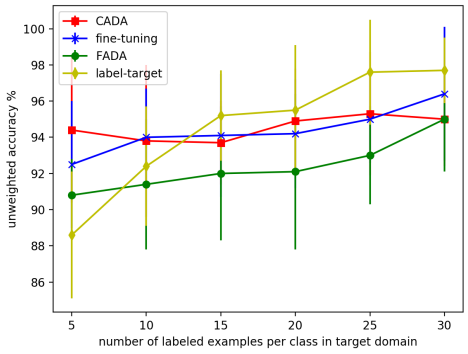
(b) Aibo (ang hap)



(c) IEMOCAP (ang hap)



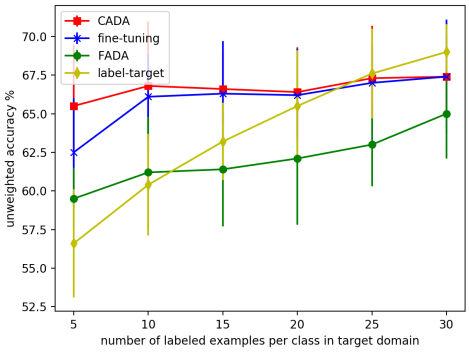
(d) IEMOCAP (ang sad)



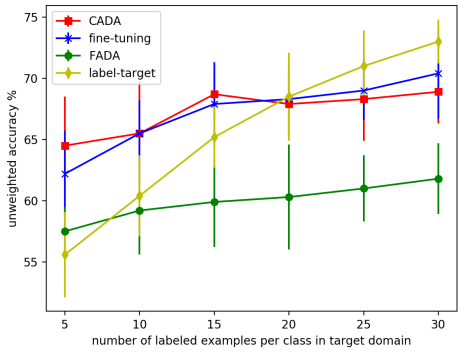
(e) IEMOCAP (sad hap)

Figure 6.1: Comparisons for the **cross-corpora basic-emotion binary-class** tasks using EMODB as target domain (source domain and emotion classes seen in the subfigure title) with three domain adaptation methods and the baseline *label-target*.

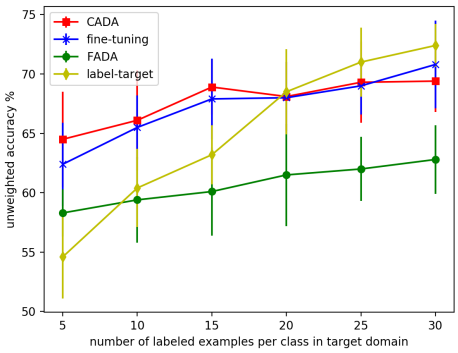
- It is surprising to find that even only very few examples (e.g. 2 per class) are available for domain adaptation, the improvement on performance is evident.



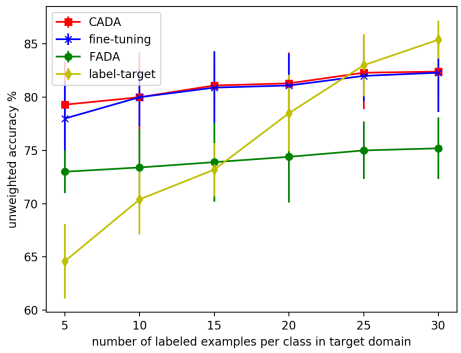
(a) Ohm (ang hap)



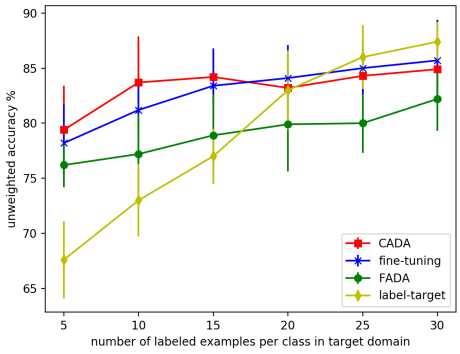
(b) Aibo (ang hap)



(c) IEMOCAP (ang hap)



(d) IEMOCAP (ang sad)



(e) IEMOCAP (sad hap)

Figure 6.2: Comparison for the **cross-corpora basic-emotion binary-class** tasks using SAVEE as target domain (source domain and emotion classes seen in the sub-figure title) with three domain adaptation methods and the baseline *label-target*.

- The dataset Aibo contains Ohm, but the results of using Aibo and Ohm for the same tasks are only slightly different. This suggests that more diversity in the

source domain does not guarantee a better adaptation performance as more diversity may lead to larger domain shift.

- Large standard deviation (between 3.1 and 7.2) is due to the randomness of the used examples in the target domain. This is understandable because some examples can be more representative of the true target-domain distribution than others. For the same reason, using more target examples can achieves slightly worse performance occasionally.

Further statistical t-test shows that the accuracy of mean by CADA is significantly different to the mean by other methods over the 20 trials of experiments with p value generally less than 0.05, as seen in the tables. The trend of change in accuracy by different methods with the target examples increasing is also illustrated in Figure 6.1 and Figure 6.2 . Particularly, the baseline *label-target* (using only labelled target data for training) is provided. From Figure 6.1-6.2, it can be viewed that

- All domain adaptation methods benefit from using more target-domain examples, though the progress can be slow compared to the baseline *label-target*, which stands for traditional supervised learning and is severely limited when the training data lacks. As seen from the figures, the trend line (in yellow) representing *label-target* usually starts at a low position (with a low accuracy) but grows quickly with the data increasing.

- CADA performs best when the target examples are few, and fine-tuning as well as the baseline *label-target* will gain an advantage over CADA when more target examples are used to tune the source model. Supposing that the target examples are sufficient for adaptation, we infer that fine-tuning can be a highly practical method.

- Among the three domain adaptation methods, FADA cannot rival others in performance and is also less sensitive to the change of target data size for adaptation.

- The performance of domain adaptation methods will basically reach stable (when the target data per class has about 20 examples, as shown in the figures) after a stage of increase. In fact, the learning algorithm in CADA and FADA decides that a balance will be finally reached between the source and target domains (by contrast, *label-target* only involves the target domain and has no such issue). On the other hand, fine-tuning can be more inclined to the target domain when the

Table 6.7: Hyper-parameter choices and accuracy of source models for the cross-corpora basic-emotion multi-class tasks

Domain	Emotions	Hyper-parameters	Accuracy %
Ohm	ang hap neu	neurons 128; batch 32; epochs 200	47.3 ± 8.6
Aibo	ang hap neu	neurons 128; batch 32; epochs 500	22.3 ± 9.0
IEMOCAP	ang hap sad	neurons 128; batch 64; epochs 300	46.2 ± 11.4
IEMOCAP	ang hap sad neu	neurons 64; batch 32; epochs 200	21.4 ± 10.5

data for tuning has a considerable size, and this process as well as the outcome is more controllable by human operators. As illustrated in the figures, the accuracy by fine-tuning can be higher than that by CADA at the end when the number of target data reaches about 30, and the advantage should be further expanded with more target data.

6.2.2 Basic-emotion multi-class tasks

Setting tasks

Following the same principles as stated in the binary-class experiment, the tasks for multi-class tasks are set as follows (the first dataset is the source and the second dataset is the target, with emotion classes in brackets)

- Ohm-EMODB/SAVEE (ang hap neu)
- Aibo-EMODB/SAVEE (ang hap neu)
- IEMOCAP-EMODB/SAVEE (ang sad hap)
- IEMOCAP-EMODB/SAVEE (ang sad hap neu)

Model selection

Similarly as in the binary-class experiment, model selection is performed and the results are given in Table 6.7. From Table 6.7, we can find that with MLP modelling, the accuracy of some tasks are rather low (lower than random guess, in the second and fourth task). Interestingly, for the same 3 emotion classes (ang hap neu), there is a big difference in accuracy between using Ohm and using Aibo, considering Ohm is contained in Aibo. On the other hand, the difference is small when only 2 classes (ang hap) are estimated. This suggests neutral emotional state can be highly difficult to recognize when mixed with other prototype emotions, and its distribution in the dataset

Table 6.8: Unweighted accuracy (%) when using EMODB as target domain in **cross-corpora basic-emotion multi-class** experiment. The numbers in the head row represent the amount of used target-domain examples per class for adaptation. P-value of t-test is also provided to ensure the difference of the top two larger means of accuracy by the three methods is on a significant level.

source	scheme	2	4	6	8	10	15	20
Ohm	FT	59.5	62.1	63.2	65.6	**68.2**	**69.5**	69.0
(ang, hap, neu)	FADA	59.3	61.8	63.1	66.6	67.2	69.3	69.1
all-source: 56.2	CADA	**59.8**	**64.7**	**65.6**	**67.1**	67.2	67.0	**69.4**
t-test	p-value	0.04	0.04	0.03	0.04	0.01	0.06	0.07
IEMOCAP	FT	67.8	70.4	71.7	73.0	**73.6**	74.5	**74.6**
(ang, hap, sad)	FADA	67.9	70.2	71.5	72.9	73.0	74.1	74.5
all-source: 65.3	CADA	**69.4**	**71.4**	**72.7**	**73.4**	73.0	**74.7**	74.3
t-test	p-value	0.09	0.03	0.04	0.02	0.08	0.14	0.07

Ohm and Mont can be very different even these two datasets are collected under similar settings with the same annotation scheme. Such observation seems consistent with our experiences as neutral states are often subtle and fuzzy compared to other distinctive emotional states. Unsurprisingly, in the dataset IEMOCAP, when neutral becomes one of the emotion classes, the performance is also very poor.

Based on the discussion, the tasks listed above will be adjusted, and only Ohm (with emotions ang hap neu) and IEMOCAP (with emotions ang sad hap) are preserved as the source domain.

Result analysis

Table 6.8 and Table 6.9 reports the results. From the tables we can basically draw the same conclusions as found in the binary-class experiments: 1) using more target data is useful for all the adaptation methods; 2) CADA is most advantageous adaptation approach, especially when target data is very few, but the advantage decreases with more target data being added for training. Such observation can also be made based on Figure 6.3 which depicts the change trend of UA (unweighted accuracy) with increasing number of target data for adaptation.

Table 6.9: Unweighted accuracy (%) when using SAVEE as target domain in **cross-corpora basic-emotion multi-class** experiment. The numbers in the head row represent the amount of used target-domain examples per class for adaptation. P-value of t-test is also provided to ensure the difference of the top two larger means of accuracy by the three methods is on a significant level.

source	scheme	2	4	6	8	10	15	20
Ohm	FT	54.0	55.2	58.1	59.7	61.2	**62.8**	**63.7**
(ang, hap, neu)	FADA	52.8	54.0	58.5	59.5	61.4	62.6	62.4
all-source: 52.9	CADA	**54.1**	**56.6**	**60.2**	**60.4**	**63.6**	62.2	63.3
t-test	p-value	0.03	0.06	0.02	0.05	0.01	0.03	0.09
IEMOCAP	FT	56.9	58.7	62.9	63.6	64.0	64.2	**65.8**
(ang, hap, sad)	FADA	55.2	58.1	60.3	62.7	63.4	64.3	63.7
all-source: 54.2	CADA	**57.3**	**60.1**	**64.3**	**63.9**	**64.0**	**64.8**	65.7
t-test	p-value	0.02	0.04	0.07	0.02	0.04	0.08	0.06

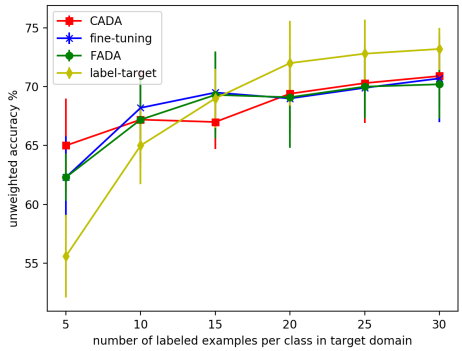
Table 6.10: Emotion classes into Positive/Negative categories

Dataset	Positive	Negative
EMODB	happy, neutral,	anger, fear, boredom, sad, disgust
SAVEE	happy, surprise, neutral	anger, disgust, fear, sadness
Ohm/Mont	all others	anger, touchy, emphatic, reprimanding
IEMOCAP	happy, excited, neutral	anger, sad, frustrated

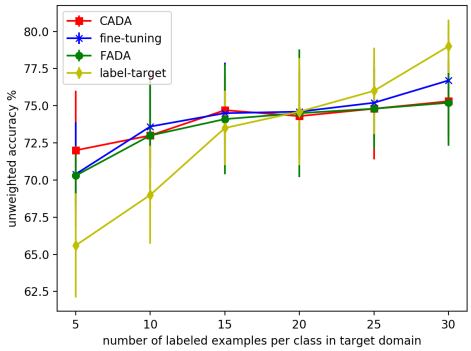
6.2.3 General-emotion tasks

Setting tasks

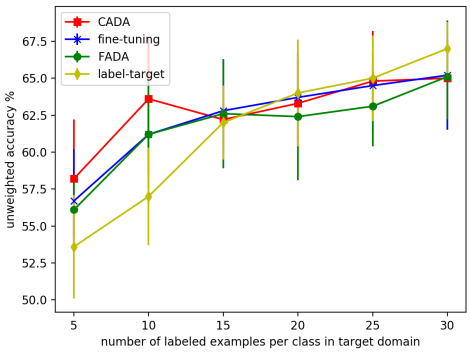
Under this setting we use the general emotion class labels. Particularly we classify all the original emotion classes in the dataset as Positive or Negative, as shown in Table 6.10. Note that neutral is also taken as belonging to the category of Positive in our setting. It is consistent with the general experiences to use an information-rich domain as the source, so we take Ohm, Mont, or IEMOCAP as the source corpus, and EMODB or SAVEE as the target corpus. The size of the data is presented in Table 6.11. Because the data size is larger due to combining different classes, the influence of class imbalance in training has been less significant. The results of model selection is given in Table 6.12, and it shows that the accuracy between Positive and Negative class is acceptable.



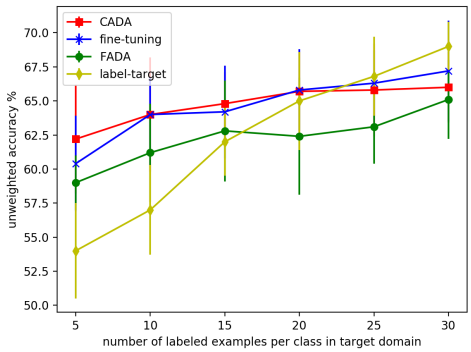
(a) Ohm-EMODB (ang hap neu)



(b) IEMOCAP-EMODB (ang hap sad)



(c) Ohm-SAVEE (ang hap neu)



(d) IEMOCAP-SAVEE (ang hap sad)

Figure 6.3: Comparison for the **cross-corpora basic-emotion multi-class** tasks (source and target domains and emotion classes seen in the sub-figure title) with three domain adaptation methods and the baseline *label-target*.

Results analysis

Table 6.13 and Table 6.14 report the performance when very few examples in the target domain are used. For briefness we omit the specific standard deviation values in the tables. The standard deviation values are around 3%-5% for most cases of CADA and fine-tuning. They are smaller for FADA (about 2%), which can be explained by FADA being not sensitive to the change of target examples and the results mainly depending on the source data. Due to the nature of supervised domain adaptation, the effectiveness of adaptation highly depends on the informativeness of the labelled target data which are used in the training process. That can explain the relatively large standard deviation for CADA and fine-tuning. From the tables, we can see that all the

Table 6.11: Size of Positive and Negative categories

Dataset	Positive	Negative
EMODB	150	385
SAVEE	240	240
Ohm	6601	3358
Mont	5792	2465
IEMOCAP	3344	4036

Table 6.12: Hyper-parameters choices and accuracy of source models for cross-corpora general-emotion (Positive/Negative) tasks

Domain	Hyper-parameters	Accuracy %
Ohm	neurons 256; batch 32; epochs 300	79.8 ± 5.1
Mont	neurons 256; batch 32; epochs 300	76.7 ± 5.3
IEMOCAP	neurons 128; batch 32; epochs 200	69.5 ± 5.8

adaptation methods achieve better performance than the baseline *all-source*, suggesting that domain shift adaptation is effective even when the target data is very limited. Among the three schemes, CADA is most advantageous while FADA performs worst, which is consistent with the analysis based the toy dataset in Chapter 5. Moreover, it is clear that adaptation brings in more improvement on the target corpus EMODB than on SAVEE. This is unsurprising as EMODB itself can be easier to perform classification. Besides, different source-target domain pairs demonstrate different domain shift conditions: the shift between the source corpora and EMODB can be simpler to address than that between the same corpora and SAVEE as suggested by the results.

It is interesting to observe the effect of domain adaptation with more target data added. Figure 6.4 shows the performance of different adaptation methods and the MLP with the baseline *label-target* by using different numbers of target label data. Notice that three large corpora as the source and two small corpora as the target domain form a total of 6 combinations. It is evident from Figure 6.4 that the performance of those adaptation methods is better than that of *label-target* setting when there are a few labelled data in the target domain. In particular, our CADA always outperforms other adaption methods until there are sufficient target data for the *label-target* baseline.

6.3 Intra-corpus experiment

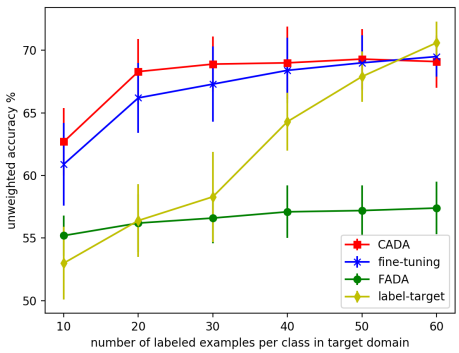
The experiments under the cross-corpora setting involve totally different databases. How about the different speakers from one corpus? Is domain adaption still useful and

Table 6.13: Unweighted accuracy (%) when using EMODB as source domain for **cross-corpora general-emotion** tasks. The numbers in the head row represent the amount of used target-domain examples per class for adaptation. P-value of t-test is also provided to ensure the difference of the top two larger means of accuracy by the three methods is on a significant level.

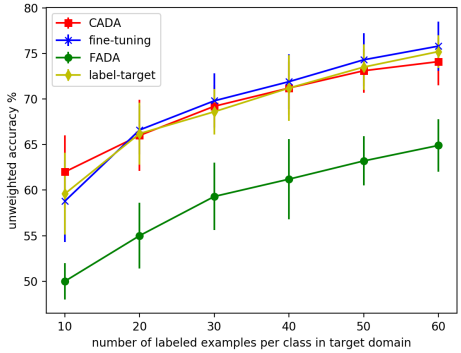
source	scheme	2	4	6	8	10	15	20
Ohm *all-source*: 55.2	FT	55.2	57.3	59.0	60.4	60.9	62.5	66.2
	FADA	54.2	54.3	54.3	54.9	55.2	55.9	56.2
	CADA	**58.1**	**59.2**	**61.8**	**62.9**	**62.7**	**65.8**	**68.2**
t-test	p-value	0.02	0.07	0.06	0.04	0.01	0.04	0.05
Mont *all-source*: 51.3	FT	55.0	55.2	56.9	58.2	59.1	63.4	64.9
	FADA	50.0	50.2	51.4	51.9	52.0	53.1	54.3
	CADA	**57.1**	**59.4**	**60.2**	**60.1**	**60.9**	**64.3**	**67.2**
t-test	p-value	0.06	0.16	0.22	0.06	0.05	0.07	0.08
IEMOCAP *all-source*: 49.2	FT	54.1	57.2	**59.2**	60.0	60.2	61.3	62.1
	FADA	49.9	50.0	50.4	50.5	50.9	51.2	53.4
	CADA	**55.2**	**58.3**	58.8	**60.2**	**61.4**	**64.3**	**66.1**
t-test	p-value	0.09	0.04	0.07	0.02	0.05	0.04	0.03

Table 6.14: Unweighted accuracy (%) when using SAVEE as source domain for **cross-corpora general-emotion** tasks. The numbers in the head row represent the amount of used target-domain examples per class for adaptation. P-value of t-test is also provided to ensure the difference of the top two larger means of accuracy by the three methods is on a significant level.

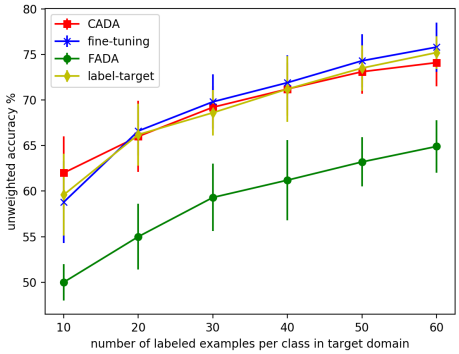
source	scheme	2	4	6	8	10	15	20
Ohm *all-source*: 53.4	FT	53.3	53.4	54.2	54.3	54.2	54.4	54.9
	FADA	53.1	53.2	54.3	54.4	54.3	54.2	54.1
	CADA	**54.1**	**54.9**	**55.2**	**56.3**	**56.5**	**57.1**	**58.2**
t-test	p-value	0.05	0.04	0.12	0.11	0.01	0.07	0.06
Mont *all-source*: 50.3	FT	52.1	52.2	53.0	53.4	53.5	54.8	55.1
	FADA	50.5	50.9	51.0	51.7	52.8	52.9	54.1
	CADA	**54.3**	**54.4**	**54.5**	**54.5**	**54.5**	**55.2**	**56.4**
t-test	p-value	0.06	0.04	0.05	0.02	0.01	0.05	0.06
IEMOCAP *all-source*: 52.1	FT	53.1	53.2	53.4	53.8	54.5	56.2	56.3
	FADA	51.9	53.0	53.2	53.3	53.3	53.4	53.4
	CADA	**53.5**	**54.2**	**54.4**	**55.0**	**56.1**	**56.3**	**56.9**
t-test	p-value	0.04	0.04	0.06	0.02	0.02	0.08	0.09



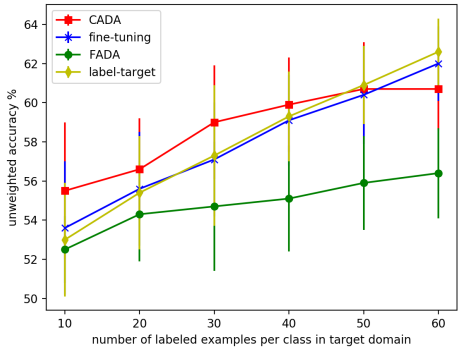
(a) Ohm-EMODB



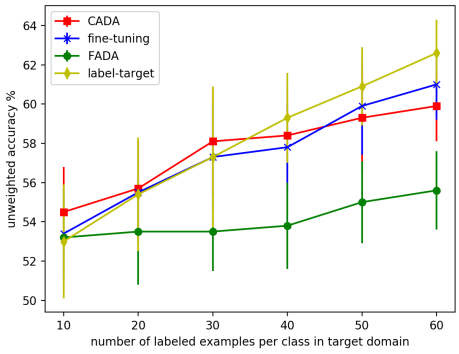
(b) Mont-EMODB



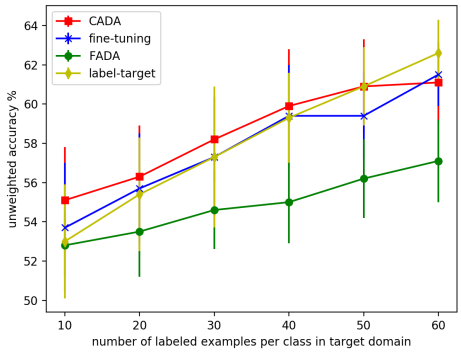
(c) IEMOCAP-EMODB



(d) Ohm-SAVEE



(e) Mont-SAVEE



(f) IEMOCAP-SAVEE

Figure 6.4: Comparison of domain adaptation methods for the **cross-corpora general-emotion** tasks (source and target domains seen in the sub-figure title) with three domain adaptation methods and the baseline *label-target*.

whether CADA is still the most advantageous approach? As discussed in Chapter 4, domain shift also occurs within one corpus as different speakers are involved. With

Table 6.15: Intra-corpus (IEMOCAP) tasks

Task	Class	Size
2-class	Positive	3344
	Negative	4036
3-class	Positive	1636
	Negative	4036
	Neutral	1708
5-class	Anger	1103
	Sadness	1084
	Happiness	595
	Frustration	1849
	Excitement	1041

the database IEMOCAP which consists of 5 relatively even sessions, we could answer these questions. Moreover, thanks to the same emotion class definitions and annotation methods, we also design the three-class and five-class tasks in order to observe the effect of CADA in multi-class problems. Details about these class re-categorizations are given in Table 6.15.

Within the corpus IEMOCAP, because the five sessions of the data are collected following the same principles and annotated with the same method, we can consider both the speaker-dependent and speaker-independent settings. The main difference between these two settings is that under the speaker-dependent setting, some utterances of the testing speakers may emerge in the training stage (the test utterances are unknown in the training stage), while under the speaker-independent setting, no utterances of the testing speakers are available in training. The speaker-independent setting is regarded by some researchers [94, 92] as able to provide more reliable evaluation for speech emotion recognition.

For intra-corpus tasks, they are formed based on the sessions defined in IEMOCAP. There are in total 5 independent sessions, and thus we can set some sessions as source domain and the remaining as target domain. Particularly, under speaker-dependent setting we use 3 sessions as source and others as target, leading to 10 combinations. Under speaker-independent setting, three sessions are used as source, one session as adaptation set for training, and one session as testing set, leading to 20 combinations. Data size of the five sessions are provided in Table 6.16. Model selection is performed for 2-class, 3-class, and 5-class tasks respectively, and the result is given in Table 6.17. Basically, the source model accuracy gets lower with more classes added.

Table 6.16: Data size of the five sessions in IEMOCAP

	Session 1	Session 2	Session 3	Session 4	Session	$\sum$
anger	229	137	240	327	170	1103
sadness	194	197	305	143	245	1084
happiness	135	117	135	65	143	595
frustration	280	325	382	481	381	1849
excitedness	143	210	151	238	299	1041
neutral	384	362	320	258	384	1708

Table 6.17: Hyper-parameters choices and accuracy of source models for intra-corpus tasks

Classes	Hyper-parameters	Accuracy %
2	neurons 256; batch 32; epochs 100	75.5 ± 3.8
3	neurons 256; batch 32; epochs 200	60.3 ± 5.7
5	neurons 256; batch 32; epochs 300	56.7 ± 5.6

6.3.1 Speaker-dependent setting

The results of using a random 10% of the target data for adaptation are reported in Tables 6.18-6.20. The numbers in the first column of the tables refer to the sessions used as source and target domains. For example, the entry 123-45 indicates that Session 1, 2, and 3 in IEMOCAP are used as the source domain, and Session 4 and 5 as the target domain. The standard deviation of 20 trials for all the methods is around 1% and thus omitted in the tables for better readability. From Tables 6.18-6.20, we can tell that CADA gains an obvious advantage, suggesting that a small number of target data can be very helpful to optimize the performance on the target domain with CADA. FADA is behind CADA and it is slightly better than *all-source* (no adaptation). It is surprising to find that fine-tuning achieves the lowest accuracy and the performance after fine-tuning gets even worse than that without any tuning. The reason may be that fine-tuning leads to over-fitting on the few target examples and the adapted model drops significantly in its generalization ability. In general, the three-class task is more difficult than the binary-task as the neutral state is chosen as a separate class. We conduct t-test on these three tasks over the 10 combinations of cases and check if the mean accuracy is significantly different between the two best performers. The results are presented in Table 6.21, which suggests the advantage of CADA is more significant in case of three-class and five-class tasks with p-value 0.012 and 0.027 respectively.

Table 6.18: Results under the **speaker-dependent** setting with IEMOCAP on the **binary-class** task. Case index in the first column refers to the Sessions used as source and target domains respectively (separated by '-').

Case index	*all-source*	FT	FADA	CADA
123-45	64.2	61.4	64.3	**64.5**
124-35	62.5	59.5	**63.2**	62.9
125-34	60.5	58.3	61.4	**61.5**
134-25	63.9	63.1	64.6	**65.2**
135-24	63.7	61.4	63.3	**64.9**
145-23	60.1	59.9	60.5	**62.2**
234-15	63.6	62.4	63.7	**65.8**
235-14	61.5	61.6	62.8	**64.1**
245-13	60.6	60.0	61.3	**62.8**
345-12	64.9	62.2	64.4	**66.1**
average	62.5	60.1	63.0	**64.0**

Table 6.19: Results under the **speaker-dependent** setting with IEMOCAP on the **three-class** task. Case index in the first column refers to the Sessions used as source and target domains respectively (separated by '-').

Case index	*all-source*	FT	FADA	CADA
123-45	50.8	48.0	48.2	**52.8**
124-35	49.0	46.3	48.4	**51.0**
125-34	47.0	43.7	46.2	**49.6**
134-25	51.6	50.3	51.6	**54.2**
135-24	50.7	47.5	51.7	**52.9**
145-23	47.3	47.0	51.6	**51.7**
234-15	51.1	48.9	51.6	**54.3**
235-14	47.1	47.2	51.5	**52.2**
245-13	47.1	46.1	51.7	**51.8**
345-12	51.1	50.2	51.6	**55.5**
average	49.3	47.5	50.4	**52.6**

Table 6.20: Comparison of domain adaptation approaches under the **speaker-dependent** setting with IEMOCAP on the **five-class** task. Case index in the first column refers to the Sessions used as source and target domains respectively (separated by '-').

Case index	*all-source*	FT	FADA	CADA
123-45	44.4	41.5	41.3	**44.8**
124-35	**43.7**	40.7	43.3	43.4
125-34	43.7	40.9	43.5	**45.0**
134-25	44.3	41.9	42.8	**44.6**
135-24	43.1	42.1	42.4	**45.6**
145-23	43.1	41.9	42.6	**44.8**
234-15	47.0	43.1	46.1	**47.2**
235-14	44.5	43.0	45.4	**47.6**
245-13	44.6	42.5	44.8	**46.4**
345-12	47.2	44.9	46.5	**48.6**
average	44.6	42.3	43.9	**45.8**

Table 6.21: T-test on the difference of accuracy under speaker-dependent setting within-corpus (IEMOCAP)

Task	Comparison	p-value
2-class	FADA/CADA	0.100
3-class	FADA/CADA	0.012
5-class	*all-source*/CADA	0.027

Table 6.22: Results under the **speaker-independent** setting with IEMOCAP on the **binary-class** task. Case index in the first column refers to the Sessions used as source domain, target training set, and target testing set, respectively (separated by '-').

Case index	*all-source*	FT	FADA	CADA
123-4-5	65.2	64.6	**65.4**	64.6
124-5-3	58.0	57.7	**58.8**	58.6
125-3-4	**62.3**	61.7	62.0	61.0
134-2-5	63.6	64.7	**65.8**	65.5
135-4-2	**65.8**	65.6	63.9	65.1
145-3-2	64.2	65.4	64.1	**66.5**
234-5-1	60.2	**63.3**	62.9	62.9
235-1-4	61.8	61.8	**62.2**	61.5
245-1-3	**60.6**	59.5	59.6	56.7
345-2-1	**63.6**	62.9	61.1	61.9
123-5-4	64.2	61.4	**64.3**	63.5
124-5-3	62.5	59.6	**63.2**	62.9
125-4-3	60.7	58.3	61.4	**61.5**
134-5-2	63.1	63.1	64.6	**65.2**
135-4-2	63.7	62.4	**64.3**	62.9
145-3-2	57.1	**57.9**	56.5	57.2
234-5-1	63.6	61.4	63.7	**65.8**
235-4-1	61.6	61.6	62.8	**64.1**
245-3-1	60.6	62.0	**63.3**	62.8
345-2-1	64.9	62.2	64.4	**66.1**
average	62.5	61.9	62.7	**63.1**

6.3.2 Speaker-independent setting

The composition of IEMOCAP allows us to choose some sessions as source data and the rest as target data. To ensure a speaker-independent setting, three of the five sessions are selected for training, and one for adaptation, and the remaining one for testing. This gives a total of 20 combinations. Because all the sessions have the same emotion class definitions and annotation methods, we consider the three-class and five-class tasks (see Table 6.15 for re-categorization) as well. The model structure and specification are the same as used in the speaker-dependent experiments.

We present the results of binary-class, three-class, and five-class tasks in Table 6.22, Table 6.23, and Table 6.24 (standard deviation is about 1 and omitted in tables). It should be emphasized that under this speaker-independent setting, the speakers/sessions used for training, adaptation, and testing are completely separate. This may lead

Table 6.23: Results under the **speaker-independent** setting with IEMOCAP on the **three-class**task. Case index in the first column refers to the Sessions used as source domain, target training set, and target testing set, respectively (separated by '-').

Case index	*all-source*	FT	FADA	CADA
123-4-5	**52.8**	53.3	49.7	52.7
124-5-3	45.0	45.3	43.2	**46.1**
125-3-4	**48.7**	48.1	43.9	48.5
134-2-5	51.0	53.5	50.1	**54.1**
135-4-2	53.5	**54.0**	53.9	53.7
145-3-2	52.3	53.9	52.1	**55.4**
234-5-1	48.6	49.5	48.7	**50.6**
235-1-4	**48.6**	48.1	44.0	47.9
245-1-3	**47.2**	45.6	41.9	43.4
345-2-1	48.7	48.6	**49.7**	48.9
123-5-4	54.2	51.4	**54.3**	53.2
124-3-5	52.5	**56.5**	53.2	52.9
125-4-3	50.5	53.3	54.4	**54.5**
134-5-2	53.9	53.1	54.6	**55.2**
135-2-4	53.7	**56.4**	54.3	55.9
145-2-3	49.1	51.9	**52.5**	52.2
234-1-5	46.6	**47.4**	47.1	46.8
235-4-1	44.5	45.6	46.8	**47.1**
245-3-1	50.6	51.0	**51.3**	51.2
345-1-2	47.9	50.2	50.4	**51.3**
average	47.8	50.8	49.8	**51.1**

Table 6.24: Results under the **speaker-independent** setting with IEMOCAP on the **five-class** task. Case index in the first column refers to the Sessions used as source domain, target training set, and target testing set, respectively (separated by '-').

Case index	*all-source*	FT	FADA	CADA
123-4-5	42.4	**43.9**	42.6	**43.9**
124-5-3	43.4	43.5	42.9	**44.3**
125-3-4	**43.9**	43.4	41.3	42.4
134-2-5	43.7	**44.4**	43.2	**44.4**
135-4-2	44.4	45.0	42.3	**45.1**
145-3-2	43.0	**45.0**	42.9	44.9
234-5-1	46.8	**47.0**	46.3	46.5
235-1-4	42.6	**43.4**	41.0	41.7
245-1-3	43.6	**44.2**	43.2	43.3
345-2-1	46.3	**47.2**	46.6	46.5
123-5-4	41.4	42.9	43.6	**43.9**
124-3-5	44.4	44.5	44.9	**45.3**
125-4-3	**43.9**	43.4	41.3	42.4
134-5-2	43.7	43.4	43.2	**44.4**
135-2-4	44.4	45.0	42.3	**45.1**
145-2-3	47.0	49.0	**49.9**	48.9
234-1-5	46.8	**47.0**	46.3	46.5
235-4-1	42.6	**43.4**	41.0	41.7
245-3-1	**43.6**	42.2	41.4	42.3
345-1-2	42.3	43.4	45.6	**46.5**
average	44.0	**44.6**	43.6	44.5

to the situation that the session of data used for adaptation do not follow the real distribution of the target/test data. In other words, domain adaptation can not work in many cases. In fact, the domain shift between the adaptation session and the testing session can even be larger than that between the training (source) and the testing session. In this situation, domain adaptation only makes the model even less suitable to the target domain. Reflected in the tables, we can find that in many cases, the domain adaptation approaches perform worse than the baseline *all-source*. In the cases where adaptation proves useful such as in 124-5-3, 134-2-5, and 145-3-2, we can find that CADA achieves only slightly better performance than the others. This reveals that when there are appropriate data/domains for adaptation, even simple fine-tuning technique can make a significant improvement. In general, from the averaged results of the three tasks, CADA demonstrates a small advantage among the domain adaptation approaches.

6.4 Evaluation on u-CADA

We use IEMOCAP to verify the effectiveness of u-CADA. To simulate the unsupervised domain adaptation scenario, besides the source domain, 10% of the unlabelled target data are provided. We have the following methods for comparison.

- *all-source*: using only the source data for training and no adaptation is needed.
- Fine-tuning (FT): tuning the source model with genuine target data.
- u-CADA: applying CADA with the source data and pseudo-labelled data.
- CADA: using source data and the genuine labelled data (the same examples used in other methods)

We select 3 sessions of IEMOCAP as the source domain and the remaining 2 sessions as the target domain. We keep the model specification and other experiment settings the same as used before and consider binary-class, three-class, five-class tasks. The experiment results are reported in Tables 6.25-6.27. It is clearly seen that u-CADA is inferior to CADA but outperforms fine-tuning in most cases and *all-source* in some cases. This suggests pseudo-labels can also be used for CADA. However, the quality of pseudo-labels are not guaranteed to yield an effective adaptation. As a consequence, we can see from the tables that only in the case of three-class task, u-CADA

Table 6.25: **Unsupervised domain adaptation** under the speaker-dependent setting with IEMOCAP on the **binary-class** task. Case index in the first column refers to the Sessions used as source and target domains respectively (separated by '-').

Case index	*all-source*	FT	u-CADA	CADA
123-45	64.2	61.4	62.3	64.5
124-35	62.5	59.5	61.9	62.9
125-34	60.5	58.3	59.4	61.5
134-25	63.9	63.1	63.9	65.2
135-24	63.7	61.4	63.0	64.9
145-23	60.1	59.9	59.4	62.2
234-15	63.6	62.4	63.6	65.8
235-14	61.5	61.6	60.4	64.1
245-13	60.6	60.0	60.1	62.8
345-12	64.9	62.2	64.5	66.1
average	62.5	60.1	61.9	64.0

Table 6.26: **Unsupervised domain adaptation** under the speaker-dependent setting with IEMOCAP on the **three-class** task. Case index in the first column refers to the Sessions used as source and target domains respectively (separated by '-').

Case index	*all-source*	FT	u-CADA	CADA
123-45	50.8	48.0	50.1	52.8
124-35	49.0	46.3	48.8	51.0
125-34	47.0	43.7	46.6	49.6
134-25	51.6	50.3	51.1	54.2
135-24	50.7	47.5	48.9	52.9
145-23	47.3	47.0	47.2	51.7
234-15	51.1	48.9	51.7	54.3
235-14	47.1	47.2	46.5	52.2
245-13	47.1	46.1	51.4	51.8
345-12	51.1	50.2	52.2	55.5
average	49.3	47.5	49.5	52.6

Table 6.27: **Unsupervised domain adaptation** approaches under the speaker-dependent setting with IEMOCAP on the **five-class** task. Case index in the first column refers to the Sessions used as source and target domains respectively (separated by '-').

Case index	*all-source*	FT	u-CADA	CADA
123-45	44.4	41.5	41.3	44.8
124-35	43.7	40.7	43.5	43.5
125-34	43.7	40.9	42.6	45.0
134-25	44.3	41.9	44.2	44.6
135-24	43.1	42.1	43.1	45.6
145-23	43.1	41.9	41.5	44.8
234-15	47.0	43.1	45.3	47.2
235-14	44.5	43.0	44.1	47.6
245-13	44.6	42.5	43.4	46.4
345-12	47.2	44.9	46.1	48.6
average	44.6	42.3	43.5	45.8

outperforms *all-source* with a small advantage on the average. This indicates the using pseudo-labels for domain adaptation can be risky. In order to address this issue, it may be helpful to look for more effective semi-supervised learning techniques for producing higher quality pseudo-labels. We leave this for the future work.

Chapter 7

CNNs-based CADA Evaluations

Class-wise adversarial domain adaptation (CADA) can be implemented via shallow or deep neural networks with the same learning algorithm. This chapter focuses on CNNs-based evaluations with various experiment settings.

7.1 Experiment design

7.1.1 Data and tasks

The used datasets for deep models are Ohm, Mont, IEMOCAP, EMODB, and SAVEE (see Chapter 3 for details). Different from the practice in MLPs-based experiment where GeMAPs feature set is adopted for all audio clips in the datasets, the input to deep models can be raw audio clips or log-mel spectrogram according to [118]. In spite of slightly better performance with log-mel spectrogram in the situation of regular supervised learning, for brevity we use raw-audio clips as the input to 1D CNNs in this project as our objective is to estimate the application of CADA with deep models instead of pursuing the highest recognition accuracy with specific tasks..

Specifically, for data pre-processing, the sampling rate of the audio clips used is 16kHz. The length of the raw audio clips used is set 1s long. (This is relatively a short length as our experiments are limited by the computing resources available. On the other hand, 1s long audio clip is generally considered stationary in emotion expression.) If the audio clip is longer than 1s, it will be segmented to 1s long. Otherwise, it is padded to 1s long. At 16 kHz sampling rate, the audio clip can be represented as a 16000-bit vector. So, the input of 1D CNNs is the 16000-bit vectors in our experiments. The data size after processing is presented in Table 7.1. As shown in the table,

Table 7.1: Datasets used for CNNs-based evaluations

Dataset	ang	hap	sad	neu	exc
EMODB	607	328	467	337	x
SAVEE	418	428	509	805	x
Ohm	2846	2124	x	17436	x
Mont	1849	521	x	15694	x
IEMOCAP	9391	9423	4875	12440	11380

Table 7.2: The hyper-parameters range in CNN

Hyper-parameter	Range
filters	{32, 64}
kernel size	{2, 3, 4}
pooling size	{3, 4, 5}
batch size	{32, 64, 128}
output layer neurons	{64, 128, 256, 512}

the data size has greatly expanded in contrast to the original data (audio clips) as each audio clip is usually 4-5s long and thus can be divided into 4 or 5 examples.

7.1.2 Model architecture and selection

We learn from [118] about the basic hyper-parameters set in CNNs for speech emotion recognition. We use 1D CNN as the basic model which consists of 2 convolutional layers and 1 fully connected layer for output, as illustrated in Figure 7.1. We adopt batch normalization, max pooling, and use elu as activation function (softmax for output layer). Optimizer is stochastic gradient descent (SGD) and loss function is categorical cross-entropy. Other hyper-parameters are selected based on 5-fold cross validation and the range is shown in Table 7.2. With the model selection, in the convolutional layer, we set filters 64, kernel size 3, and stride 1. For the pooling layer, we set pooling size 4 and stride 4. The batch size is 64 and neuron number in the fully-connected layer is 128.

For class-wise adversarial learning, the two convolutional layers (as well as the pooling layers) are treated as the feature encoder, and the fully connected layer as the predictor. Both parts are updated following the learning rules in Algorithm 1 (Chapter 5). The implementation uses TensorFlow (Keras library) on python, and considering the need of powerful computing systems, we use cloud computing platform Amazon EC2 to run the experiment.

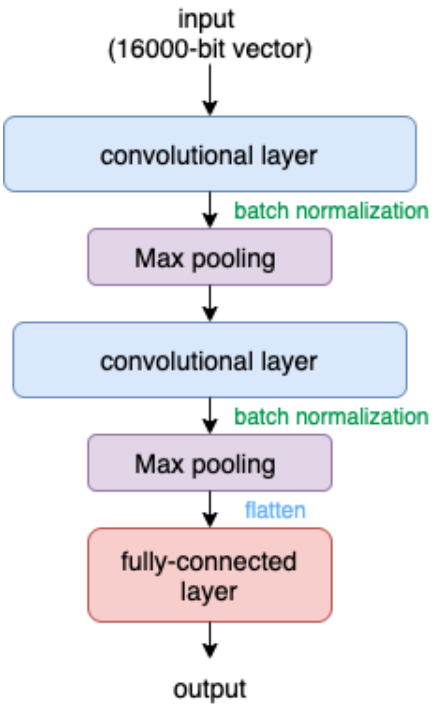


Figure 7.1: The 1D CNN model

7.1.3 Baselines and comparative approaches

The baseline as presented in the MLPs-CADA experiment, is *all-source* that predicts the target data with the source model without any adaptation skills. As for domain adaptation methods, FADA is not used as the comparative approach, as the computing capacity it requires for deep models is demanding to us. Consequently we only consider fine-tuning as the comparative domain adaptation approach. The results of random 20 trials with the same experiment settings are reported.

7.2 Cross-corpora experiment

As designed in the MLPs-based evaluations, we have cross-corpora experiments first, using large datasets as source domains and small datasets as target domains. We divide the tasks based on whether they are binary-class or multi-class.

7.2.1 Binary-class tasks

Model selection on the source domain suggests the hyper-parameters which are used in building CADA. Regarding domain adaptation, we randomly select a small part of

Table 7.3: Unweighted accuracy (%) when using EMODB as target domain in **cross-corpora basic-emotion binary-clas**s experiment based CNNs. The numbers in the head row represent the percent of the used target data for training/adaptation. P-value of t-test is also provided to ensure the difference of the two means of accuracy by the two methods is on a significant level.

source	scheme	1%	2%	4%	6%	8%	10%	20%
Ohm (ang, hap)	FT	54.6	55.7	57.6	58.4	58.7	58.9	**59.6**
all-source: 53.2	CADA	**55.8**	**57.4**	**58.6**	**59.3**	**59.8**	**59.1**	59.1
t-test	p-value	0.03	0.04	0.02	0.02	0.01	0.04	0.06
Aibo (ang, hap)	FT	52.1	53.2	55.1	**56.4**	**57.0**	**58.2**	**59.0**
all-source: 51.8	CADA	**53.5**	**54.0**	**55.4**	56.2	56.5	57.1	58.4
t-test	p-value	0.02	0.04	0.05	0.02	0.08	0.04	0.07
IEMOCAP (ang, hap)	FT	63.3	63.7	64.2	64.8	**67.2**	69.0	**68.5**
all-source: 62.0	CADA	**63.6**	**66.2**	**67.8**	**69.0**	67.1	**69.4**	67.5
t-test	p-value	0.02	0.04	0.01	0.07	0.06	0.08	0.06
IEMOCAP (ang, sad)	FT	86.7	88.2	90.3	92.1	**94.6**	**95.0**	**98.1**
all-source: 85.2	CADA	**88.4**	**89.4**	**92.5**	**92.8**	93.0	94.1	96.9
t-test	p-value	0.03	0.04	0.02	0.02	0.01	0.05	0.06
IEMOCAP (sad, hap)	FT	82.1	83.4	83.9	84.8	**88.0**	**91.1**	92.4
all-source: 80.6	CADA	**83.1**	**84.3**	**85.4**	**87.6**	87.9	90.7	**92.8**
t-test	p-value	0.04	0.09	0.12	0.02	0.11	0.09	0.05

Table 7.4: Unweighted accuracy (%) when using SAVEE as target domain in **cross-corpora basic-emotion binary-class** experiment based on CNNs. The numbers in the head row represent the percent of the used target data for training/adaptation. P-value of t-test is also provided to ensure the difference of the two means of accuracy by the two methods is on a significant level.

source	scheme	1%	2%	4%	6%	8%	10%	20%
Ohm (ang, hap)	FT	55.1	55.3	54.7	55.6	56.1	56.3	58.2
all-source: 54.3	CADA	**56.9**	**57.6**	**56.4**	**57.0**	**57.4**	**58.6**	**59.4**
t-test	p-value	0.04	0.04	0.07	0.02	0.07	0.06	0.08
Aibo (ang, hap)	FT	55.9	58.2	60.2	61.6	**62.5**	63.9	**65.3**
all-source: 53.7	CADA	**57.2**	**60.2**	**62.6**	**63.4**	**62.5**	**65.7**	64.9
t-test	p-value	0.02	0.05	0.01	0.02	0.07	0.04	0.09
IEMOCAP (ang, hap)	FT	57.9	61.2	64.4	**64.3**	65.5	67.9	68.0
all-source: 57.6	CADA	**59.8**	**62.9**	**66.8**	63.5	**66.1**	**68.9**	**68.1**
t-test	p-value	0.03	0.06	0.07	0.02	0.01	0.09	0.12
IEMOCAP (ang, sad)	FT	63.2	67.8	68.9	69.4	**70.0**	70.9	71.1
all-source: 62.1	CADA	**65.9**	**69.2**	**69.8**	**70.2**	**70.0**	**71.6**	**71.3**
t-test	p-value	0.07	0.08	0.03	0.02	0.01	0.09	0.06
IEMOCAP (sad, hap)	FT	65.3	67.8	69.1	70.3	71.4	73.5	**74.1**
all-source: 65.0	CADA	**66.3**	**69.0**	**71.9**	**72.4**	**73.6**	**74.2**	73.3
t-test	p-value	0.04	0.10	0.06	0.04	0.01	0.05	0.06

Table 7.5: Unweighted accuracy (%) when using EMODB as target domain in **cross-corpora basic-emotion multi-class** experiment based on CNNs. The numbers in the head row represent the percent of the used target data for training/adaptation. P-value of t-test is also provided to ensure the difference of the two means of accuracy by the two methods is on a significant level.

source	scheme	1%	2%	4%	6%	8%	10%	20%
Ohm (ang, hap, neu)	FT	59.5	60.1	64.1	64.6	66.2	67.5	**69.1**
all-source: 56.8	CADA	**62.1**	**64.7**	**64.9**	**65.6**	**67.2**	**68.5**	68.6
t-test	p-value	0.13	0.01	0.02	0.02	0.01	0.02	0.04
IEMOCAP (ang, hap, sad)	FT	65.8	65.2	68.5	69.2	70.6	71.4	72.9
all-source: 65.2	CADA	**66.9**	**67.2**	**69.5**	**69.9**	**71.6**	**72.4**	**73.9**
t-test	p-value	0.05	0.09	0.02	0.02	0.03	0.04	0.07

target data for domain adaptation and gradually increase that proportion from 1% to 20%. The largest proportion (20%) represents 187 examples being used from a total of 935 examples in EMODB for classification between anger and happiness. By contrast, for the same task in MLPs-based experiment, 20 per class of the total of 187 examples is about 21%, similar to the proportion set here.

From Table 7.3 and Table 7.4, it is viewed that both domain adaptation approaches outperform the baseline in almost all of the cases, verifying the value of domain adaptation. Particularly, CADA, in most cases, shows a better performance than fine-tuning, especially when the used target-data are few. An intuitive explanation is that because CNNs are more sophisticated architectures involving much more parameter than MLPs, when only a few target data are provided, fine-tuning may have little effect on modifying those huge number of parameters, thus possibly achieving a similar or only slightly better performance compared to the model without adaptation. However, with more target data for tuning, the adapted model will show a significant improvement and even outperform CADA, in some cases, at the point of 10% target data being used as the tables suggest.

Another observation we have is that the standard deviation of the accuracy is generally smaller compared to that in the MLPs-based experiment. Basically, the range of standard deviation is [2, 4] while for MLPs is [3, 8]. This indicates CNNs are more stable in training, and the learned features in the model are more robust.

Table 7.6: Unweighted accuracy (%) when using SAVEE as target domain in **cross-corpora basic-emotion multi-class** experiment based on CNNs. The numbers in the head row represent the percent of the used target data for training/adaptation. P-value of t-test is also provided to ensure the difference of the two means of accuracy by the two methods is on a significant level.

source	scheme	1%	2%	4%	6%	8%	10%	20%
Ohm (ang, hap, neu)	FT	54.0	55.2	57.1	58.7	60.2	**64.8**	**65.7**
all-source: 52.9	CADA	**55.8**	**57.2**	**59.1**	**60.7**	**62.2**	**64.8**	64.7
t-test	p-value	0.02	0.05	0.02	0.02	0.10	0.09	0.08
IEMOCAP (ang, hap, sad)	FT	61.9	64.7	68.3	**69.6**	71.3	72.3	73.7
all-source: 60.8	CADA	**63.2**	**65.7**	**68.6**	**69.6**	**73.4**	**74.3**	**75.7**
t-test	p-value	0.05	0.04	0.02	0.02	0.02	0.03	0.06

7.2.2 Multi-class tasks

For multiple class task, the experiment results are shown in Table 7.5 and Table 7.6. The source domains and emotion classes for classification are seen in the tables (the tasks are set following the practice in the corresponding experiments based on MLPs.).

We have similar observations as in the binary class tasks. With more emotion classes in the task, the recognition performance does not necessarily get worse as some emotions can be easier to recognize and thus may increase the overall recognition accuracy, which is calculated on all the classes. Between Table 7.5 and Table 7.6, we can find SAVEE is easier to address as the target domain than EMODB, as indicated by the *all-source* performance. However, with domain adaptation, the recognition accuracy with both datasets can reach a similar level, especially when using IEMOCAP as the source domain. This suggests the domain adaptation effect is more significant on SAVEE under this setting.

7.3 Intra-corpus experiment

IEMOCAP is used to conduct intra-corpus experiments by selecting certain sessions of the dataset as source domain and the rest as target domain. We also consider both speaker-dependent and speaker-independent settings. Specifically, among the five sessions of IEMOCAP, three sessions are used as source, the other two are used as one target domain (which will be further split for adaptation and testing), or in case of speaker-independent setting, as one adaptation set and one test set. Details of the five sessions is given in Table 7.7, which shows that there is relatively sufficient data for

Table 7.7: Data size in IEMOCAP for CNN-based evaluations

	Session 1	Session 2	Session 3	Session 4	Session 5
ang	2247	1117	2068	2512	1447
exc	1568	1828	1522	1843	2662
hap	1099	860	1084	693	1139
neu	2664	3041	2277	1752	2705
sad	2134	2207	2691	1796	2548
$\sum$	9712	9053	9642	8596	10501

Table 7.8: Results under the **speaker-dependent** setting with IEMOCAP on the **binary-class** task based on CNNs. Case index in the first column refers to the Sessions used as source and target domains respectively (separated by '-').

Case index	*all-source*	FT	CADA
123-45	61.2	61.4	**64.5**
124-35	58.5	59.8	61.9
125-34	55.5	56.3	**58.5**
134-25	61.2	61.3	**64.2**
135-24	63.7	62.4	**64.4**
145-23	60.2	59.1	**62.2**
234-15	59.6	62.4	62.7
235-14	64.3	64.6	**66.1**
245-13	62.7	**65.0**	64.8
345-12	61.9	61.3	**64.1**
average	60.9	61.1	**63.4**

each section, favoring the training of deep models.

7.3.1 Speaker-dependent setting

We basically follow the same setting as in the MLPs-based experiment. Emotion classes are re-grouped to form binary-class, and three-class and five-class tasks. The results are given in Table 7.8 - 7.10. It can be viewed from these tables that domain adaptation approaches achieves a better performance than the baseline *all-source*. Meanwhile, CADA performs better than fine-tuning in most cases. This observation is consistent with what we have found in MLPs-based evaluations.

We respectively conduct t-test for these three tasks over the 10 combinations of cases (reported in Table 7.11), comparing the accuracy by CADA, fine-tuning, or *all-source* (two larger means among the three are compared), verifying the advantage of CADA in the tasks on a significant level.

Table 7.9: Results under the **speaker-dependent** setting with IEMOCAP on the **three-class** task based on CNNs. Case index in the first column refers to the Sessions used as source and target domains respectively (separated by '-').

Case index	*all-source*	FT	CADA
123-45	50.8	48.0	**52.8**
124-35	49.0	46.3	**51.0**
125-34	47.0	43.7	**49.6**
134-25	51.6	50.3	**54.2**
135-24	50.7	47.5	**52.9**
145-23	47.3	47.0	**51.7**
234-15	51.1	48.9	**54.3**
235-14	47.1	47.2	**52.2**
245-13	47.1	46.1	**51.8**
345-12	51.1	50.2	**54.5**
average	49.2	47.5	**52.5**

Table 7.10: Comparison of domain adaptation approaches under the **speaker-dependent** setting with IEMOCAP on the **five-class** task based on CNNs. Case index in the first column refers to the Sessions used as source and target domains respectively (separated by '-').

Case index	*all-source*	FT	CADA
123-45	53.4	54.5	**58.8**
124-35	**53.5**	52.0	53.4
125-34	47.1	49.9	**52.3**
134-25	42.3	41.9	**45.7**
135-24	48.1	**49.1**	50.5
145-23	52.1	52.6	**56.8**
234-15	45.0	43.1	**48.2**
235-14	44.2	43.0	**48.6**
245-13	47.0	47.1	**51.4**
345-12	45.0	44.9	**48.9**
average	47.7	47.5	**51.5**

Table 7.11: T-test on the difference of accuracy under speaker-dependent setting within-corpus (IEMOCAP) with CNNs.

Task	Comparison	p-value
2-class	FT/CADA	0.058
3-class	*all-source*/CADA	0.0005
5-class	*all-source*/CADA	0.053

Table 7.12: Results under the **speaker-independent** setting with IEMOCAP on the **binary-class** task based on CNNs. Case index in the first column refers to the Sessions used as source domain, target training set, and target testing set, respectively (separated by '-').

Case index	*all-source*	FT	CADA
123-4-5	55.2	54.3	54.6
124-5-3	58.0	57.7	58.6
125-3-4	**62.3**	61.7	61.0
134-2-5	53.8	54.7	55.5
135-4-2	64.8	65.2	**66.1**
145-3-2	64.2	65.4	**66.5**
234-5-1	60.2	**63.3**	62.9
235-1-4	61.8	61.8	61.5
245-1-3	**60.6**	59.5	58.7
345-2-1	**63.6**	62.9	61.9
123-5-4	64.2	61.4	63.5
124-5-3	62.5	59.6	62.9
125-4-3	60.7	58.3	**61.5**
134-5-2	63.1	63.1	**65.2**
135-4-2	58.7	59.4	**61.2**
145-3-2	57.1	**57.9**	57.2
234-5-1	57.6	58.4	**60.8**
235-4-1	61.6	**64.6**	64.2
245-3-1	60.6	62.0	62.8
345-2-1	61.9	**62.1**	61.3
average	60.6	60.7	61.4

7.3.2 Speaker-independent setting

Regarding the speaker-independent setting, similar to previous practice, we address the total of 20 combinations of IEMOCAP sessions. The results about 2-class, 3-class, and 5-class tasks are given in Table 7.12 - 7.14. As shown in the first column of the tables, the digits signify the number sessions used as source data, adaptation data, and test data. From the tables, we can find that

- The results generally confirm our hypothesis that different sessions demonstrate significant domain shift, and the data used for adaptation may not be representative of the the test data set (this hypothesis already verified in the MLPs-based experiment in Chapter 6). As a consequence, no domain adaptation approaches can achieve an overall improvement over the baseline; and there is not a winner

Table 7.13: Results under the **speaker-independent** setting with IEMOCAP on the **three-class** task based on CNNs. Case index in the first column refers to the Sessions used as source domain, target training set, and target testing set, respectively (separated by '-').

Case index	*all-source*	FT	CADA
123-4-5	54.3	55.6	55.4
124-5-3	55.0	55.3	**58.1**
125-3-4	**48.7**	48.1	48.5
134-2-5	51.0	53.5	**54.1**
135-4-2	53.5	**54.0**	53.7
145-3-2	52.3	56.9	**58.4**
234-5-1	48.6	49.5	50.6
235-1-4	**48.6**	48.1	47.9
245-1-3	**47.2**	45.6	43.4
345-2-1	48.7	48.6	**49.6**
123-5-4	54.2	51.4	53.2
124-3-5	53.5	**55.5**	52.9
125-4-3	50.5	53.3	**54.5**
134-5-2	53.9	**57.1**	56.4
135-2-4	53.7	54.4	**55.9**
145-2-3	49.1	49.9	**52.2**
234-1-5	46.6	**47.4**	45.8
235-4-1	54.3	55.7	**58.1**
245-3-1	50.6	51.0	51.2
345-1-2	50.9	**54.2**	**54.2**
average	51.3	52.3	**52.5**

Table 7.14: Results under the **speaker-independent** setting with IEMOCAP on the **five-class** task based on CNNs. Case index in the first column refers to the Sessions used as source domain, target training set, and target testing set, respectively (separated by '-').

Case index	*all-source*	FT	CADA
123-4-5	42.4	**43.9**	**43.9**
124-5-3	43.4	43.5	**46.3**
125-3-4	**49.4**	48.4	43.9
134-2-5	43.7	44.4	**48.7**
135-4-2	44.4	**45.2**	45.1
145-3-2	43.0	**45.0**	44.9
234-5-1	**47.0**	46.8	46.2
235-1-4	42.7	**43.4**	41.7
245-1-3	42.6	43.2	**43.3**
345-2-1	46.3	47.2	46.5
123-5-4	51.7	54.8	54.6
124-3-5	43.4	44.5	45.6
125-4-3	**43.9**	43.4	42.4
134-5-2	**47.4**	44.4	43.7
135-2-4	42.4	**45.3**	45.1
145-2-3	49.1	50.6	**52.9**
234-1-5	46.8	**47.0**	46.5
235-4-1	43.1	45.4	**48.7**
245-3-1	42.3	41.2	42.3
345-1-2	42.0	46.4	**47.5**
average	44.7	45.7	**46.2**

among the two adaptation approaches (although on average, CADA looks better but the advantage is minor).

- Interestingly, by comparing the results with those generated in MLPs-based experiments under the same setting (Table 6.22-6.24), there are obviously more cases showing domain adaptation performs better than the baseline *all-source* (no adaptation). This indicates the adapted deep models demonstrate better generalization capability than MLPs (thus more suitable on unknown testing examples), thanks to the robust feature representations embedded in the convolutional layers which are learned in the adversarial learning.

7.4 Discussion

It should be emphasized that it is not comparable between MLPs-based and CNN-based experiments with respect to the performance (unweighted accuracy). The main reasons are that the input in CNNs is 16000-bit vector, which stands for a 1s long audio clip, while in MLPs the input is 62-bit vector, which stands for the extracted 62 features from the whole audio clip (usually longer than 1s). The data for testing are also in the same form as the input, and that makes the comparison of accuracy between them infeasible.

However, it is worth finding that applying CADA to deep neural networks is workable, yielding better performance than regular fine-tuning technique (certainly better than no-adaptation practice) on the target domain. In addition, the experiments within-corpus (IEMOCAP) under speaker-independent setting suggests deep models show better generalization capability due to robust feature representations learned by convolutional layers. This inspires us to explore more deep learning models for speech emotion recognition in the future towards building more universal and practical recognition systems.

In summary, this chapter presents how to use CADA for deep models. Our goal is to assessing the effectiveness of CADA rather than pursue highest recognition accuracy, which can be achieved by using more complex architectures and longer input audios.

Chapter 8

Conclusions

This thesis explores domain adaptation via adversarial learning for speech emotion recognition. Our work is originally motivated by the well-known cross-corpora problem and the issue of data scarcity in speech emotion data. Data is the foundation of machine learning models, and a lack of emotion data makes it difficult to build up a robust recognition system. Besides collecting more large-scale high-quality databases, a promising solution to data scarcity is domain adaptation that utilises a related but information-rich domain to help address the target domain. The key to the success of domain adaptation is to eliminate the domain shift, the divergence of data distributions across different domains. However, the nature of speech emotion data determines that the domain shift can be very complex. High variability can occur not only across corpora but also within one corpus or one certain emotion class.

We list and analyze the factors that may contribute to the domain shift in speech emotion data. Given two different corpora, recognizing these factors can be rather straightforward, and it is unsurprising that even within one corpus, testing a model under the speaker-dependent and speaker-independent setting can make a big difference.

When considering the supervised domain adaptation, we suppose that annotating a few examples sometimes can be a better solution than collecting many unlabelled data. Then how to make best use of these limited information is a crucial step. The state-of-the-art supervised domain adaptation approach FADA adopts pairing technique to tackle the scarcity of the target data, while we find that class-wise adversarial domain adaptation (CADA) works even when only a few target data are available. Pairing technique also uses the emotion class information when generating the pairs, but CADA explicitly perform adversarial learning for each common class, and thus can cope with more complex intra-class variability. To evaluate the proposed method, we

have considered both toy datasets and real-world datasets, binary-class and multi-class recognition, simple model and deep model, and the speaker-dependent and speaker-independent settings. The speaker-independent setting in our context of domain adaptation means that speakers whose utterances used for adaptation will not emerge in the testing stage, and it is observed from our experiments that domain shift cannot be effectively eliminated under this situation.

Implementing CADA is straightforward as well as extending it to the unsupervised domain adaptation scenario. We find that using pseudo-labels generated by the model trained on source domain can be beneficial for adaptation. Yet the performance is not guaranteed, as it depends heavily on the target data distribution and its similarity with the used source domain.

Restricted by the author's capability and the time/computing resources available for the project, there are some issues not covered or addressed adequately. Some specific issues include

- For CNN-based CADA, we use raw audio clips as input to deep models. Another popular way worth trying is using log-mel spectrogram, which can be superior to audio clips in training deep models with supervised learning. It is meaningful to find out whether this judgement holds in the scenario of domain adaptation.

- The length of examples we use in deep models is relatively short, and it may cause some global information missing. Using examples containing more information should improve the overall performance (also require more powerful computing systems).

- Although we have conducted MLPs-based and CNNs-based evaluations, we cannot compare the results directly because that in MLPs, the examples for testing is an integral audio clip (turn) which usually contains 4-5 times more information than a single example used in CNNs. A potential solution is to trim the raw audio clips to a fixed length, and then convert the clip to the input vectors to CNNs.

In general our work can be extended and refined with the following aspects.

- The datasets we use in this work are limited and some popular latest databases are not covered. Although we have considered various evaluation skills, the proposed approach should be tested with more natural emotional corpora under even more challenging experiment settings.

- Our work only uses CNNs with two convolutional layers while more deep architectures and other types of deep models like long-short time memory (LSTM) can be more suitable to process speech data.

- Multi-modal emotion recognition is found to be more competitive than single-modal emotion recognition, and is attracting more attention. It is also our interest to explore the combination of body language, ECG, facial expression, and speech, for more robust and reliable emotion recognition systems.

- Regarding unsupervised domain adaptation, although we propose u-CADA as a variant of CADA for that scenario, the pseudo labels we leverage are produced by the source model, which in itself, is not appropriate for predicting target data because of domain shift. Therefore more sophisticated pseudo-labelling technique should be utilised and semi-supervised learning may provide some insights to this.

- Our work only considers the categorical/discrete emotion theories. Investigating the application of CADA for continuous theories, which are also widely applied in the area of affective computing, is an important direction.

In summary, this thesis focuses on supervised domain adaptation, a practical scenario in speech emotion recognition as collecting emotion samples are often difficult, and in assessing the current adaptation approaches, it is found that the high intra-class variability existing in emotion data limits the adversarial learning from eliminating the caused domain shift. To cope with this issue, class-wise adversarial domain adaptation (CADA) is proposed. CADA is characterized by combining the two main components, feature encoder and prediction layer into one structure, leading to very straightforward learning rules and relatively simple implementation. A comprehensive evaluation proves the effectiveness and efficiency of CADA under different contexts.

Bibliography

[1] Mohammed Abdelwahab and Carlos Busso. Supervised domain adaptation for emotion recognition from speech. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 5058–5062. IEEE, 2015.

[2] Christos-Nikolaos Anagnostopoulos, Theodoros Iliou, and Ioannis Giannoukos. Features and classifiers for emotion recognition from speech: a survey from 2000 to 2011. *Artificial Intelligence Review*, 43(2):155–177, 2015.

[3] Anton Batliner, Stefan Steidl, and Elmar Nöth. Releasing a thoroughly annotated and processed spontaneous emotional database: the fau aibo emotion corpus. 2008.

[4] Yoann Baveye, Jean-Noël Bettinelli, Emmanuel Dellandréa, Liming Chen, and Christel Chamaret. A large video database for computational models of induced emotion. In *2013 Humaine Association Conference on Affective Computing and Intelligent Interaction*, pages 13–18. IEEE, 2013.

[5] Pascal Belin, Sarah Fillion-Bilodeau, and Frédéric Gosselin. The montreal affective voices: a validated set of nonverbal affect bursts for research on auditory affective processing. *Behavior research methods*, 40(2):531–539, 2008.

[6] John Blitzer, Mark Dredze, and Fernando Pereira. Domain adaptation for sentiment classification. In *45th Annv. Meeting of the Assoc. Computational Linguistics (ACL'07)*.

[7] Cynthia Breazeal and Lijin Aryananda. Recognition of affective communicative intent in robot-directed speech. *Autonomous robots*, 12(1):83–104, 2002.

[8] Yuheng Bu, Shaofeng Zou, Yingbin Liang, and Venugopal V Veeravalli. Estimation of kl divergence: Optimal minimax rate. *IEEE Transactions on Information Theory*, 64(4):2648–2674, 2018.

[9] Felix Burkhardt, Astrid Paeschke, Miriam Rolfes, Walter F Sendlmeier, and Benjamin Weiss. A database of german emotional speech. In *Interspeech*, volume 5, pages 1517–1520, 2005.

[10] Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N Chang, Sungbok Lee, and Shrikanth S Narayanan. Iemocap: Interactive emotional dyadic motion capture database. *Language resources and evaluation*, 42(4):335, 2008.

[11] Houwei Cao, Ragini Verma, and Ani Nenkova. Speaker-sensitive emotion recognition via ranking: Studies on acted and spontaneous speech. *Computer speech & language*, 29(1):186–202, 2015.

[12] Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

[13] Girish Chandrashekar and Ferat Sahin. A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1):16–28, 2014.

[14] Louis C Charland. Emotion as a natural kind: Towards a computational foundation for emotion theory. *Philosophical psychology*, 8(1):59–84, 1995.

[15] Roddy Cowie, Ellen Douglas-Cowie, Nicolas Tsapatsoulis, George Votsis, Stefanos Kollias, Winfried Fellenz, and John G Taylor. Emotion recognition in human-computer interaction. *IEEE Signal processing magazine*, 18(1):32–80, 2001.

[16] Darwin. *The expression of the emotions in man and animals*. University of Chicago press, 2015.

[17] Jun Deng, Rui Xia, Zixing Zhang, Yang Liu, and Björn Schuller. Introducing shared-hidden-layer autoencoders for transfer learning and their application in acoustic emotion recognition. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4818–4822. IEEE, 2014.

[18] Jun Deng, Zixing Zhang, Florian Eyben, and Björn Schuller. Autoencoder-based unsupervised domain adaptation for speech emotion recognition. *IEEE Signal Processing Letters*, 21(9):1068–1072, 2014.

[19] Jun Deng, Zixing Zhang, Erik Marchi, and Björn Schuller. Sparse autoencoder-based feature transfer learning for speech emotion recognition. In *Affective Computing and Intelligent Interaction (ACII), 2013 Humaine Association Conference on*, pages 511–516. IEEE, 2013.

[20] Anton Dries and Ulrich Rückert. Adaptive concept drift detection. *Statistical Analysis and Data Mining*, 2(5-6):311–327, 2009.

[21] P Eckman. Universal and cultural differences in facial expression of emotion. In *Nebraska symposium on motivation*, volume 19, pages 207–284. University of Nebraska Press Lincoln, 1972.

[22] Moataz El Ayadi, Mohamed S Kamel, and Fakhri Karray. Survey on speech emotion recognition: Features, classification schemes, and databases. *Pattern Recognition*, 44(3):572–587, 2011.

[23] Florian Eyben, Anton Batliner, Björn Schuller, Dino Seppi, Stefan Steidl, et al. Cross-corpus classification of realistic emotions–some pilot experiments. In *Proc. LREC workshop on Emotion Corpora, Valettea, Malta*, pages 77–82, 2010.

[24] Florian Eyben, Klaus R Scherer, Björn W Schuller, Johan Sundberg, Elisabeth André, Carlos Busso, Laurence Y Devillers, Julien Epps, Petri Laukka, Shrikanth S Narayanan, et al. The geneva minimalistic acoustic parameter set (gemaps) for voice research and affective computing. *IEEE Transactions on Affective Computing*, 7(2):190–202, 2016.

[25] Haytham M Fayek, Margaret Lech, and Lawrence Cavedon. Evaluating deep learning architectures for speech emotion recognition. *Neural Networks*, 92:60–68, 2017.

[26] Daniel Joseph France, Richard G Shiavi, Stephen Silverman, Marilyn Silverman, and M Wilkes. Acoustical properties of speech as indicators of depression and suicidal risk. *IEEE transactions on Biomedical Engineering*, 47(7):829–837, 2000.

[27] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, pages 1180–1189, 2015.

[28] Nestor Garay, Idoia Cearreta, Juan López, and Inmaculada Fajardo. Assistive technology and affective mediation. *Human technology*, 2(1):55–83, 2006.

[29] Daniel Garcia-Romero and Carol Y Espy-Wilson. Analysis of i-vector length normalization in speaker recognition systems. In *Interspeech*, pages 249–252, 2011.

[30] Kiel Gilleade, Alan Dix, and Jen Allanson. Affective videogames and modes of affective gaming: assist me, challenge me, emote me. *DiGRA 2005: Changing Views–Worlds in Play.*, 2005.

[31] Milan Gnjatovic and Dietmar Rosner. Inducing genuine emotions in simulated speech-based human-machine interaction: The nimitek corpus. *IEEE Transactions on Affective Computing*, 1(2):132–144, 2010.

[32] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[33] Mark K Greenwald, Edwin W Cook, and Peter J Lang. Affective judgment and psychophysiological response: Dimensional covariation in the evaluation of pictorial stimuli. *Journal of psychophysiology*, 3(1):51–64, 1989.

[34] Michael Grimm, Kristian Kroschel, and Shrikanth Narayanan. The vera am mittag german audio-visual emotional speech database. In *2008 IEEE international conference on multimedia and expo*, pages 865–868. IEEE, 2008.

[35] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.

[36] Steven Handel. Classification of emotions, 2012.

[37] John HL Hansen, Sahar E Bou-Ghazale, Ruhi Sarikaya, and Bryan Pellom. Getting started with susas: a speech under simulated and actual stress database. In *Eurospeech*, volume 97, pages 1743–46, 1997.

[38] Ali Hassan, Robert Damper, and Mahesan Niranjan. On acoustic emotion recognition: compensating for covariate shift. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(7):1458–1468, 2013.

[39] Robert Hecht-Nielsen. Theory of the backpropagation neural network. In *Neural networks for perception*, pages 65–93. Elsevier, 1992.

[40] David R Heise. Enculturating agents with expressive role behavior. *Agent culture: Human-agent interaction in a multicultural world*, pages 127–142, 2004.

[41] Magnus Rudolph Hestenes, Eduard Stiefel, et al. *Methods of conjugate gradients for solving linear systems*, volume 49. NBS Washington, DC, 1952.

[42] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

[43] Stefan Hoch, Frank Althoff, Gregor McGlaun, and Gerhard Rigoll. Bimodal fusion of emotional data in an automotive environment. In *Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, volume 2, pages ii–1085. IEEE, 2005.

[44] Yuhuang Hu, Adrian Huber, Jithendar Anumula, and Shih-Chii Liu. Overcoming the vanishing gradient problem in plain recurrent networks. *arXiv preprint arXiv:1801.06105*, 2018.

[45] Xuedong Huang, Alex Acero, Hsiao-Wuen Hon, and Raj Foreword By-Reddy. *Spoken language processing: A guide to theory, algorithm, and system development*. Prentice hall PTR, 2001.

[46] David Hume. *A treatise of human nature*. Courier Corporation, 2003.

[47] David Hume. Emotions and moods. *Organizational behavior*, pages 258–297, 2012.

[48] P Jackson and S Haq. Surrey audio-visual expressed emotion (savee) database. *University of Surrey: Guildford, UK*, 2014.

[49] Qiang Ji, Peilin Lan, and Carl Looney. A probabilistic framework for modeling and real-time monitoring human fatigue. *IEEE Transactions on systems, man, and cybernetics-Part A: Systems and humans*, 36(5):862–875, 2006.

[50] Slobodan T Jovicic, Zorka Kasic, Miodrag Dordevic, and Mirjana Rajkovic. Serbian emotional speech database: design, processing and evaluation. In *9th Conference Speech and Computer*, 2004.

[51] Takafumi Kanamori, Shohei Hido, and Masashi Sugiyama. A least-squares approach to direct importance estimation. *Journal of Machine Learning Research*, 10(Jul):1391–1445, 2009.

[52] Ashish Kapoor, Winslow Burleson, and Rosalind W Picard. Automatic prediction of frustration. *International journal of human-computer studies*, 65(8):724–736, 2007.

[53] Mitu Khandaker. Designing affective video games to support the social-emotional development of teenagers with autism spectrum disorders. *Annual Review of Cybertherapy and Telemedicine*, 7:37–39, 2009.

[54] Tomi Kinnunen and Haizhou Li. An overview of text-independent speaker recognition: From features to supervectors. *Speech communication*, 52(1):12–40, 2010.

[55] Shashidhar G Koolagudi and K Sreenivasa Rao. Emotion recognition from speech: a review. *International journal of speech technology*, 15(2):99–117, 2012.

[56] Theodoros Kostoulas, Todor Ganchev, Alexandros Lazaridis, and Nikos Fakotakis. Enhancing emotion recognition from speech through feature selection. In *International Conference on Text, Speech and Dialogue*, pages 338–344. Springer, 2010.

[57] Zoltán Kövecses. *Metaphor and emotion: Language, culture, and body in human feeling*. Cambridge University Press, 2003.

[58] LB Krithika and Lakshmi Priya GG. Student emotion recognition system (sers) for e-learning improvement based on learner concentration metric. *Procedia Computer Science*, 85:767–776, 2016.

[59] George Lakoff. Language and emotion. *Emotion Review*, 8(3):269–273, 2016.

[60] Siddique Latif, Rajib Rana, Shahzad Younis, Junaid Qadir, and Julien Epps. Cross corpus speech emotion classification-an effective transfer learning technique. *arXiv preprint arXiv:1801.06353*, 2018.

[61] Chul Min Lee and Shrikanth S Narayanan. Toward detecting emotions in spoken dialogs. *IEEE transactions on speech and audio processing*, 13(2):293–303, 2005.

[62] Chul Min Lee, Serdar Yildirim, Murtaza Bulut, Abe Kazemzadeh, Carlos Busso, Zhigang Deng, Sungbok Lee, and Shrikanth Narayanan. Emotion recognition based on phoneme classes. In *Eighth International Conference on Spoken Language Processing*, 2004.

[63] Iulia Lefter, Leon JM Rothkrantz, Pascal Wiggers, and David A Van Leeuwen. Emotion recognition from speech by combining databases and fusion of classifiers. In *International Conference on Text, Speech and Dialogue*, pages 353–360. Springer, 2010.

[64] Elizabeth D Liddy. Natural language processing. 2001.

[65] Yi-Lin Lin and Gang Wei. Speech emotion recognition based on hmm and svm. In *2005 international conference on machine learning and cybernetics*, volume 8, pages 4898–4901. IEEE, 2005.

[66] Dong C Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1):503–528, 1989.

[67] Pengfei Liu, Xipeng Qiu, Xinchi Chen, Shiyu Wu, and Xuan-Jing Huang. Multi-timescale long short-term memory neural network for modelling sentences and documents. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 2326–2335, 2015.

[68] Stacy Marsella, Jonathan Gratch, Paolo Petta, et al. Computational models of emotion. *A Blueprint for Affective Computing-A sourcebook and manual*, 11(1):21–46, 2010.

[69] Leena Mary. *Extraction and representation of prosody for speaker, speech and language recognition*. Springer Science & Business Media, 2011.

[70] Saeid Motiian, Quinn Jones, Seyed Iranmanesh, and Gianfranco Doretto. Few-shot adversarial domain adaptation. In *Advances in Neural Information Processing Systems*, pages 6673–6683, 2017.

[71] Ryohei Nakatsu, Joy Nicholson, and Naoko Tosa. Emotion recognition and its application to computer agents with spontaneous interactive capabilities. *Knowledge-Based Systems*, 13(7):497–504, 2000.

[72] Paula M Niedenthal and François Ric. *Psychology of emotion*. Psychology Press, 2017.

[73] Tin Lay Nwe, Say Wei Foo, and Liyanage C De Silva. Speech emotion recognition using hidden markov models. *Speech communication*, 41(4):603–623, 2003.

[74] Keith Oatley, Dacher Keltner, and Jennifer M Jenkins. *Understanding emotions*. Blackwell publishing, 2006.

[75] Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015.

[76] Frank J Owens. *Signal processing of speech*. Macmillan International Higher Education, 1993.

[77] Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210, 2011.

[78] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

[79] Jaak Panksepp. *Affective neuroscience: The foundations of human and animal emotions*. Oxford university press, 2004.

[80] Rosalind W Picard. *Affective computing*. MIT press, 2000.

[81] Robert Plutchik. Emotions in the practice of psychotherapy-clinical implications of affect theories. 2000.

[82] Joaquin Quiñonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and N Lawrence. Covariate shift and local learning by distribution matching, 2008.

[83] Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. The MIT Press, 2009.

[84] Lawrence Rabiner and Biing-Hwang Juang. Fundamentals of speech recognition. 1993.

[85] Lawrence R Rabiner and Ronald W Schafer. *Digital processing of speech signals*. Prentice Hall, 1978.

[86] Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer, and Andrew Y Ng. Self-taught learning: transfer learning from unlabeled data. In *Proceedings of the 24th international conference on Machine learning*, pages 759–766, 2007.

[87] Fabien Ringeval, Björn Schuller, Michel Valstar, Roddy Cowie, Heysem Kaya, Maximilian Schmitt, Shahin Amiriparian, Nicholas Cummins, Denis Lalanne, Adrien Michaud, et al. Avec 2018 workshop and challenge: Bipolar disorder and cross-cultural affect recognition. In *Proceedings of the 2018 on Audio/Visual Emotion Challenge and Workshop*, pages 3–13. ACM, 2018.

[88] Fabien Ringeval, Andreas Sonderegger, Juergen Sauer, and Denis Lalanne. Introducing the recola multimodal corpus of remote collaborative and affective interactions. In *2013 10th IEEE international conference and workshops on automatic face and gesture recognition (FG)*, pages 1–8. IEEE, 2013.

[89] Matías Roodschild, Jorge Gotay Sardiñas, and Adrián Will. A new approach for the vanishing gradient problem on sigmoid activation. *Progress in Artificial Intelligence*, 9(4):351–360, 2020.

[90] Daniel Schacter, Daniel Gilbert, Daniel Wegner, and Bruce M Hood. *Psychology: European Edition*. Macmillan International Higher Education, 2011.

[91] Klaus R Scherer. What are emotions? and how can they be measured? *Social science information*, 44(4):695–729, 2005.

[92] Björn Schuller, Anton Batliner, Stefan Steidl, and Dino Seppi. Recognising realistic emotions and affect in speech: State of the art and lessons learnt from the first challenge. *Speech Communication*, 53(9):1062–1087, 2011.

[93] Björn Schuller, Stefan Steidl, Anton Batliner, Felix Burkhardt, Laurence Devillers, Christian MüLler, and Shrikanth Narayanan. Paralinguistics in speech

and language—state-of-the-art and the challenge. *Computer Speech & Language*, 27(1):4–39, 2013.

[94] Bjorn Schuller, Bogdan Vlasenko, Florian Eyben, Martin Wollmer, Andre Stuhlsatz, Andreas Wendemuth, and Gerhard Rigoll. Cross-corpus acoustic emotion recognition: variances and strategies. *IEEE Transactions on Affective Computing*, 1(2):119–131, 2010.

[95] Björn Schuller, Zixing Zhang, Felix Weninger, and Gerhard Rigoll. Using multiple databases for training in emotion recognition: To unite or to vote? In *INTERSPEECH*, pages 1553–1556, 2011.

[96] Björn W Schuller. Speech emotion recognition: Two decades in a nutshell, benchmarks, and ongoing trends. *Communications of the ACM*, 61(5):90–99, 2018.

[97] Björn W Schuller, Anton Batliner, Dino Seppi, Stefan Steidl, Thurid Vogt, Johannes Wagner, Laurence Devillers, Laurence Vidrascu, Noam Amir, Loic Kessous, et al. The relevance of feature type for the automatic classification of emotional user states: low level descriptors and functionals. In *Interspeech*, pages 2253–2256, 2007.

[98] Björn W Schuller, Stefan Steidl, Anton Batliner, et al. The interspeech 2009 emotion challenge. In *Interspeech*, volume 2009, pages 312–315, 2009.

[99] Sagar Sharma and Simone Sharma. Activation functions in neural networks. *Towards Data Science*, 6(12):310–316, 2017.

[100] Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000.

[101] P Sibi, S Allwyn Jones, and P Siddarth. Analysis of different activation functions using back propagation neural networks. *Journal of theoretical and applied information technology*, 47(3):1264–1268, 2013.

[102] Malcolm Slaney and Gerald McRoberts. Baby ears: a recognition system for affective vocalizations. In *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, volume 2, pages 985–988. IEEE, 1998.

[103] Jonathan Sykes and Simon Brown. Affective gaming: measuring emotion through the gamepad. In *CHI'03 extended abstracts on Human factors in computing systems*, pages 732–733, 2003.

[104] Orit Taubman-Ben-Ari. The effects of positive emotion priming on self-reported reckless driving. *Accident Analysis & Prevention*, 45:718–725, 2012.

[105] Sebastian Thrun and Lorien Pratt. *Learning to learn*. Springer Science & Business Media, 2012.

[106] Leimin Tian, Johanna D Moore, and Catherine Lai. Emotion recognition in spontaneous and acted dialogues. In *2015 International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 698–704. IEEE, 2015.

[107] Pedro A Torres-Carrasquillo, Elliot Singer, Mary A Kohler, Richard J Greene, Douglas A Reynolds, and John R Deller Jr. Approaches to language identification using gaussian mixture models and shifted delta cepstral features. In *INTERSPEECH*, 2002.

[108] George Trigeorgis, Fabien Ringeval, Raymond Brueckner, Erik Marchi, Mihalis A Nicolaou, Björn Schuller, and Stefanos Zafeiriou. Adieu features? end-to-end speech emotion recognition using a deep convolutional recurrent network. In *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5200–5204. IEEE, 2016.

[109] Yuta Tsuboi, Hisashi Kashima, Shohei Hido, Steffen Bickel, and Masashi Sugiyama. Direct density ratio estimation for large-scale covariate shift adaptation. *Information and Media Technologies*, 4(2):529–546, 2009.

[110] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. *arXiv preprint arXiv:1702.05464*, 2017.

[111] Thurid Vogt and Elisabeth André. Comparing feature sets for acted and spontaneous speech in view of automatic emotion recognition. In *2005 IEEE International Conference on Multimedia and Expo*, pages 474–477. IEEE, 2005.

[112] Ying Wang, Shoufu Du, and Yongzhao Zhan. Adaptive and optimal classification of speech emotion recognition. In *2008 Fourth International Conference on Natural Computation*, volume 5, pages 407–411. IEEE, 2008.

[113] David Weenink. Speech signal processing with praat. *Haettu*, 16:2014, 2014.

[114] Yi Yao and Gianfranco Doretto. Boosting for transfer learning with multiple sources. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 1855–1862. IEEE, 2010.

[115] Richard Yonck. *Heart of the machine: Our future in a world of artificial emotional intelligence*. Arcade, 2020.

[116] William A Yost. Fundamentals of hearing: an introduction, 2001.

[117] Robert B Zajonc. Emotions. 1998.

[118] Jianfeng Zhao, Xia Mao, and Lijiang Chen. Speech emotion recognition using deep 1d & 2d cnn lstm networks. *Biomedical Signal Processing and Control*, 47:312–323, 2019.