University of Birmingham
School of Computer Science

BSc Computer Science 2019/20 - Final Year Project

Faisal IMH Alrajhi
ID: 1593979

Supervisor: Prof. Martin Russell

Automatic Emotion Recognition in Children's Speech

Abstract

Emotion recognition has numerous uses in human computer interaction systems, medical practices, military training and more. One resource to automatically recognize emotions from is acoustic speech data. A major issue in emotion recognition is the search for highly discriminate features appropriate for classification. This paper explores the use of state of the art I Vector feature vectors and deep neural network classifiers in automatic recognition of children's speech. A system was built using the PF Star Children's Speech corpus for emotion speech in English with six emotional classes; angry, joyful, motherese, emphatic, neutral and other. Applying Linear Discriminant Analysis on I Vectors and using these low dimensional I Vectors in a deep neural network exhibited performance upwards of 82% unweighted average recall. This is an improvement on the 69% originally reported by the PF Star dataset in German. Interestingly, longer than average MFCC frames (40ms+) improved performance and evidence suggests that using a scale-based definition for emotion could exhibit increased performance over the distinct equivalence classes used in this project. Data augmentation, cross lingual testing and hybrid classifier models have not been experimented with and are the ideal next step onwards from this project.

Keywords: emotion recognition, speech processing, ivectors, deep neural network

Contents

Introduction

1.1 Overview

Emotion recognition from speech is important with tasks involving Human-Machine interaction in the same way it is important to tasks involving Human-Human interaction. Communication through speech is the quickest method of communication between humans. If this speed can be achieved by AI systems, it could revolutionize human-computer interaction. However, more research is still needed to reach the revolutionary point. The current main issues in this field are related to the search for highly discriminate features and appropriate classification methods to which this project aims to contribute to.

1.2 Objective

The primary objective of this project was to build a system to automatically recognize emotions expressed in children's speech using acoustic speech data using the latest state of the art techniques of I Vectors and neural networks. In order to achieve this goal, the following steps were taken:

1. Conduct a review of current techniques and how they came to be used to understand potential optimizations or changes that can be made to improve recognition performance.
2. Gain a clear understanding of the dataset used to better understand how results can be interpreted.
3. Use and optimize detailed and efficient feature extraction and classification techniques while explaining the processes used in order to replicate and improve successful techniques.
4. Evaluate the results and contribute to the field of emotion recognition with any interesting, new or contradictory findings, furthering the field of research.

Literature Review

2.1 What is Emotion?

2.1.1 Definition

Wierzbicka argues emotions, such as anger, "can be defined in terms of universal semantic primitives" (Wierzbick, 2003). A collection of semantic primitves, such as good or want, are used to rigidly define emotions such that emotions which are similar, such as happy and sad, appear as two distinctly defined emotions. Caband argues against this categorical definition of emotions and instead defines them by way of a common currency (Caband, 2002). Caband proposes that "emotion is any mental experience" with high pleasure/displeasure, the common currency, and with high intensity. This definition places emotion in a scale rather than categories. A categorical approach to define and classify emotions is taken to simplify this project.

2.1.2 Classifying Emotion

The classification of emotions has evolved over time. Ekman devised a list of well established emotions in 1972 after conducting experiments on multiple cultures, labeling data from facial expressions. Ekman's six emotions included anger, disgust, fear, happiness, sadness and surprise. Researches such as Sabini and Silver argued this list is incomplete, missing emotions such as love (Sabini & Silver 2005). However, Ekman's contribution set a baseline for emotion classification built upon by future researchers.

Plutchik aimed to categorize emotions less rigidly, creating the now popular representation "wheel of emotion" (Plutchik 1982) seen in Figure 2.1. He hypothesized emotions could blend together to define equivalence classes of emotions. With further research, he improved the wheel into a conic figure, factoring in parameters such as intensity. The four baseline emotional pairs used by Plutchik are illustrated in the figure.

Parrot proposed emotion can be classified in a tree structure. Primary emotions encompass a set of Secondary emotions that break down to a set of Tertiary emotions. The structure removes any connection between emotions of a different category. For example, Love and Joy, and all emotions derived from them, are not connected in any way (Parrot 2001). Parrot defines emotions distinctly, like Ekman, rather than using a scale, like Plutchik. Unlike Ekman's categorization, Parrot's is more nuanced. Automatic emotion recognition tasks become simpler when using distinct definitions.

Research in determining distinct classes of emotion is still ongoing. Using reports of emotional states elicited by videos varying in emotional content, researchers (Cowen & Keltner 2017) observed 27 varieties of emotion. "Although categories are found to organize dimensional appraisals in a coherent and powerful fashion, many categories are linked by smooth gradients, contrary to discrete theories."

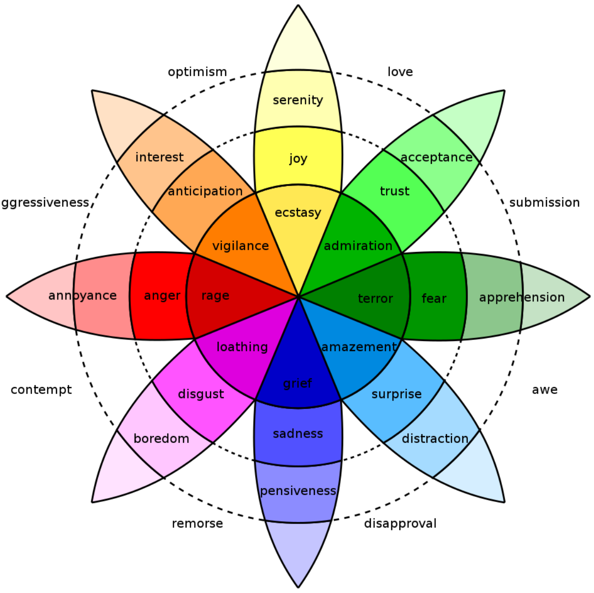

Figure 2.1: Plutchik's Wheel of Emotion

2.1.3 Applications

Automatic Emotion Recognition has many practical applications ranging from Human Computer Interaction to military training. (Landowska et. al 2014) describe uses in the domains of "software engineering, website customization, education, and gaming" in detail. Customizing websites and video games to respond to a user's emotional state can improve the user's experience by adapting to said responses. Understanding how people in education react to scenarios, such as exams, can help in improving education on an individual basis by giving a clearer profile for improvement.

A key example, monitoring soldier emotion during training provides concrete feedback when attempting to emulate real world stressful situations. Understanding the emotional engagement of soldiers in training better prepares them for the real situation they are training for. The emotional feedback can be used to reduce the impact of Post Traumatic Stress Disorder and to improve already existing PTSD therapy. (Rizzo et. al 2005)

2.1.4 Automatic Emotion Recognition

Automatic Emotion Recognition can be done using text, facial expressions and gestures, audio or a mixture of them (Seal et. al 2019). Text based recognition uses semantic labels to classify emotions. With images and videos, gestures and facial expressions are used to classify emotions. Speech data can be used as both its own set of features and as transcribed text with speech recognition to apply text-based emotion recognition. This paper aims to use acoustic speech data for automatic emotion recognition without any text based approach.

2.2 Emotion Recognition from Speech

Speech Recognition is the most common form of speech technology. However, there are more speech technologies that relay a different set of information than recognition; verification, to identify a speaker, emotion recognition, to classify emotions, and synthesis, to generate speech. Speech recognition is often concerned with the phones corresponding to letters, words and phrases and suffers in performance due to differences in the voice patterns of individuals. On the other hand, verification and identification focuses on identifying the speaker through their unique patterns. Despite our understanding of the differences, research proves features designed for one task can find similar success with other tasks (Mackova et. al 2016). Most of the signal processing and feature extraction is similar between the forms of speech technologies despite the different aims of each.

2.2.1 Data Collection

Emotion Recognition faces immediate problems from data collection and labeling. For example, Emotions are not equally defined across different cultures (Mesquita & Walker 2003). "Cultural differences exist in some aspects of emotions," so difficulties exist when perceiving emotion expressed from a person raised in a different culture than the recognizer (Lim 2016). There is evidence to the claim of emotional recognition being more universal in facial recognition and gestures (Lorette & Dewaele 2018, Matsumoto & Hwang 2019), however speech and language still suffers from cultural and language differences.

Mislabeling of emotion is not limited to cultural differences. Instances exist of humans not agreeing on identifying an emotion (Yarnell et. al 2017). While this issue is not as prevalent in a scale or currency based definition for emotion, the simpler discrete classes will produce mislabeling if not enough classes are used in labeling, as demonstrated by (Cowen & Keltner 2017).

A speech database's, or corpus's, method of collecting and labeling data is where these issues should be tackled. Each corpus aims to collect and label data authentically. When collecting emotion speech data, the emotion is ideally authentic and not acted out because if people have varying perception of emotion then their acting may not be true to the emotion expressed. Examples include the Interactive Emotional Children's Speech Corpus (IESC-Child) wherein children converse with robots in a Wizard of Oz setting to "induce different emotional reactions" authentically (Avila-George et al 2019). IESC-Child is spoken Mexican Spanish; it is important to specify the language to identify cultural implications. Similarly, the PF Star and AIBO corpora elicit natural reaction from children by placing them in a controlled scenario (Russell 2006, Batliner et. al 2004). PF Star and AIBO are in English and German, respectively, adding to the variety of languages to experiment with language differences.

2.2.2 Feature Extraction

Feature extraction is an integral process in automatic emotion recognition. Speech data features can be divided into two distinct categories; acoustic and linguistic. Acoustic features are dominantly used in speech emotion recognition (Jin et. al 2015). The usage and importance of linguistic features varies with the methods a corpus is created with; emotions acted out may not contain relevant linguistic data, but a candid corpus can provide more information to improve classifier performance (Batliner et. al 2011).

Features for audio signals are usually extracted from smaller frames within the speech signal. In short periods of time, an audio signal can be considered invariant, so traditional signal processing techniques can be used. The usual size of the frame used for speech processing is 20ms to 30ms 20ms to 30ms (Reddy & Vijayarajan 2020). Instead of taking the frames as they are, windowing functions are applied to handle the edges of the signal frame to avoid abrupt changes. Hamming and Rectangular Windowing are two common functions used (Othman & Riadh 2008).

Acoustic Features

Acoustic features can be broken down into two general categories; prosodic and spectral. As observed by (Frick 1985), "emotions can be expressed prosodically...through a variety of prosodic features." Commonly used prosodic features include pitch, energy and duration. Prosodic features are also speaker-specific (Ben Alex et. al 2018), so feature vectors extracted are generally relative rather than absolute to minimize the error from the difference.

Spectral features are derived from the frequency domain of a signal. A cepstrum, the inverse of the logarithm of a spectrum, emphasises changes in its respective spectrum. Mel frequency cepstral coefficients (MFCCs) are the most common cepstral features extracted for use in speech systems. MFCCs were originally proposed for identifying monosyllabic words in continuously spoken sentences, yet they were proven to be useful in speaker verification and emotion recognition too (Gupta 2016). MFCCs aim to replicate human hearing on the assumption the "ear is a reliable speaker recognizer" (Alim & Rashid 2017). The human ear detects differences in a smaller range of frequency clearer than larger frequency ranges. MFCCs accomplish this by applying a filters spaced linearly at low frequencies and logarithmically at high frequencies on a speech signal.

Supervectors

An issue with using MFCCs as a feature for classification algorithms, such as support vector machines, is the potential for different speech segments to have varying lengths of feature vectors. One early solution researchers used was time-averaging utterances to equalize the feature vector size (Markel et. al 1977). Despite computation efficiency, this method resulted in poorer accuracy. The common trend to improve results was to use data-to-model matching for classification, such as by using Gaussian Mixture Models (Kinnunen & Li 2010).

Gaussian Mixture Models (GMMs) require a large amount of data to be effective for an emotion class (Hu et. al 2007). A Universal Background Model (UBM) can be used instead. UBMs are class independent GMMs, so data from all emotional classes can be used to compensate for the sparse data from distinct classes. A UBM can then be adapted into a class specific GMM using an adaption algorithm based on maximum a posteriori (MAP) criterion (Chen & Chen 2016). The mean vectors of the adapted GMM form the GMM Supervector.

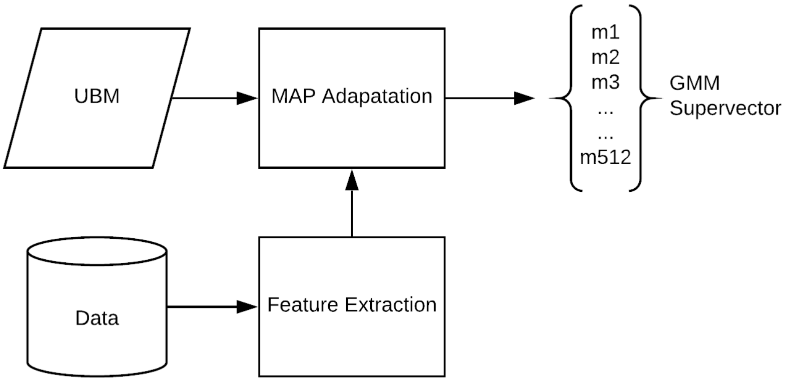

Figure 2.2: Supervector Extraction Overview

Supervectors are a single vector representation of a GMM. Utterances with feature vectors of varying length can be converted into a standard length supervector by the process illustrated in Figure 2.2. On one hand, a supervector solves both the issue of varying length features and reduced accuracy for use in classification algorithms. On the other hand, supervectors can be very high in dimension and computationally demanding.

IVectors

I Vectors are a fixed length information rich low dimensional representation of a GMM-UBM based on Join Factor Analysis techniques originally proposed by (Dehak et. al 2010). I Vectors solve the issue of computational demand from supervectors while still having a standard size; they are at the forefront of research in speaker verification and emotion recognition from speech as demonstrated by the success found in experiments in speech recognition, speaker verification and emotion recognition (Karafiat et. al 2011, Mackova et. al 2016). The success can be attributed to it providing "the benefit of modeling both the intra-domain and inter-domain variabilities into the same low dimensional space" (Ibrahim & Ramli 2018).

2.2.3 Classifiers

Automatic Emotion Recognition can be achieved using various machine learning and classification algorithms. SVMs are very commonly used for emotion recognition tasks and often exhibit the best performance (Chen & Chen 2016, Mehta et. al 2019, Casale et. al 2008). This can be attributed to an SVMs robustness to high dimensional inputs such as the supervector (Han et. al 2014). GMM-UBM models are also used as classifiers rather than feature extractors, reaching performances upwards of 79% (Schwenker et. al 2009).

Deep learning methods are used most recently. Deep neural networks (DNNs) exhibit a number of improvements; they are capable of learning high-level representations from raw features to classify data (Han et. al 2014) and can function effectively on smaller data sets. Additionally, various architectures exists capable of handling data of different types. Convolutional Neural Networks (CNNs) have remarkable performance in n-dimensional vectors, such as images or layered features (Issa et. al 2020). Recurrant Neural Networks (RNNs) “show considerable success” (Lim et. al 2014) in sequential data processing tasks, such as speech processing. Deep learning methods and hybrid neural networks have shown significant improvements in the relative range of 20% accuracy increases (Han et. al 2014) compared to previous standards.

Optimizing neural networks is required to improve the robustness and usability of an automatic emotion recognizer. (Batliner et. al 2004) achieved an overall recognition of 69.8% using a neural network using a “95 prosodic and 30 part-of-speech” features. (Xia & Liu 2016) and (Heracleous & Yoneyama 2019), and other researchers, convey the improvements of using I Vectors and neural network approach over earlier supervector and SVM techniques with their experimental results achievement results upwards of 90%/.

Design and Implementation

The main objective of the project is to build an automatic emotion recognizer from children's speech using the more recent I Vector and DNN approach. The objective can be broken down to three secondary objectives; understanding the data to be used in the project, extracting the features required for the neural network and building and optimizing the neural network. This section's structure mirrors the objective pathway.

3.1 PF Star English

The PF Star Children's Speech corpus was used in this project (Batliner et. al 2005). The corpus was designed to be cross lingual with the same experiments used to collect both German and English data. The AIBO experiment was designed to elicit and record via a microphone natural emotions from children (Batliner et. al 2004).

The experiment is described as a "Wizard-of-Oz scenario." Children were tasked with guiding a Sony AIBO robot, a dog-like robot, through a maze while fulfilling objectives. No specific instruction were given to the children besides speaking to AIBO as if it were a friend. The children were led to believe they had control over AIBO's actions, however AIBO was secretly controlled by a human operator unbeknownst to the children. Two experimental conditions were outlined; experiment E1 has the operator input instructions based on the child's commands as best as possible, emulating a high performance speech recognition system. Experiment E2, on the other hand, had the operator input a "fixed, pre-determined sequence of actions" which takes no account of the child's commands, emulating a low performance speech recognition system. The children completed both experiments, E1 and E2, in that order, and told they were using two alternative systems.

3.1.1 File Details

30 children between ages 4 and 14, inclusive, took part in both experiments. A total of 8.5 hours of recordings is provided, but only just over 1.5 hours remains once the silence, pauses and unintelligible speech are removed. The audio files were named as nX where n is a number between 1 and 31 indicating the child and x is a letter indicating the experiment, A for E1 and B or E2. Some files were missing, corrupted or lacking labels; these have been excluded from the project. The audio files contained two audio channels from different microphones; only audio from channel one was used to avoid errors that could result from mismatched audio quality.

3.1.2 Labeling Details

The corpus originally classified utterances into one of eleven emotion classes. However, due to sparsity of members in some classes, this was later revised to six emotion classes, as seen in

Table 3.1. The english data was labeled by three listeners. All three responses are available in the label files. Techniques can be used to either average, randomly select or throw away unequal labels. This project uses six emotion classes to reduce the impact of lacking data in some emotional classes. Utterances without a unanimous label are labeled randomly from the three choices. There are a total of **5302** available labeled utterances divided according to Table 3.1.

Table 3.1: Emotion classes sample weight

Emotion Class	Sample Size	Weight
Angry	326	0.062
Joyful	10	0.001
Motherese	38	0.007
Emphatic	576	0.109
Neutral	4338	0.819
Other	14	0.002

3.2 Feature Extraction

The feature vector used for classification algorithms in this project was the I Vector. Several steps, illustrated in Figure 3.1, were taken to produce these vectors. MATLAB was used to convert the audio files into usable features; additionally, the MSR Identity Toolkit by (Sadjadi et. al 2013) contained functions used for UBM, I Vectors and Linear Discriminant Analysis (LDA) calculations.

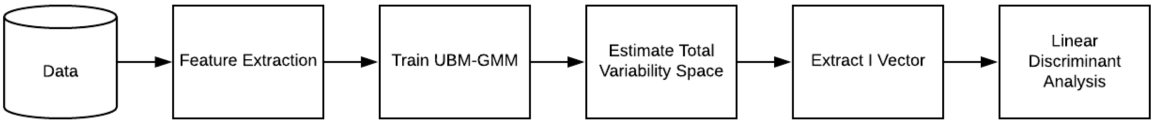
Data
Feature Extraction
Train UBM-GMM
Estimate Total Variability Space
Extract I Vector
Linear Discriminant Analysis

Figure 3.1: I Vector Extraction Steps

3.2.1 MFCC

The audio file contains all utterances for a specified child and experiment. MFCC extraction is applied to each utterance, so the audio file is divided into segments wherein each segment matches a label. The following steps are applied to each utterance to obtain an MFCC feature vector.

1. Frame the utterance into shorter frames (20ms to 30ms) with some overlap and apply a windowing function. Hamming window was used.

2. Take the Discrete Fourier Transform (DFT) of the signal to convert it into the frequency domain. This is done using the `stft(...)` function in MATLAB's Signal Processing library.
3. Apply 26 triangular filter banks to the DFT output as seen in Figure 3.2. The triangular filters match the non-linear human perception by covering a smaller area in smaller frequencies and vice versa. The log is taken of the energy in each filter.

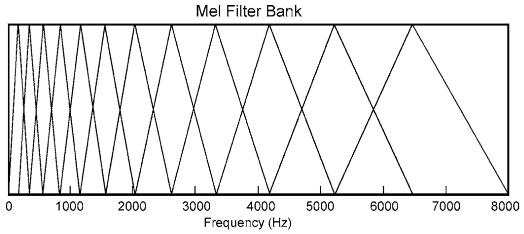

Figure 3.2: Mel Filterbanks applied to a power spectrum of a signal

4. Take the Discrete Cosine Transform (DCT) of the previous output. DCT compresses the data into, usually, 26 coefficients for the frame. The upper half are thrown out, resulting in 13 coefficients.
5. Additionally, the delta and delta-delta of the coefficients can also be calculated. MFCC coefficients contain only data from the frame analyzed; however information can also be contained with how the frames change. The delta and delta-delta are taken for this purpose.

All steps after DFT are completed in the MATLAB function `mfcc(...)`, outputting a 40 dimensional vector containing the 13 coefficients, deltas and delta-deltas and the log energy of the entire frame. The entire MFCC extraction process is done in the project using the custom `extract_mfcc(...)` function.

Following MFCC extraction, a UBM is calculated using the `gmm_em(...)` function provided in the MSR Toolkit. The UBM is created using the EM algorithm and contains a user-defined n components. This parameter is a target for optimization.

3.2.2 I Vectors

To generate I Vectors, the total variability (TV) space must first be estimated. Assuming a simple factor analysis model of the form:

$$M = m + T.x \tag{3.1}$$

where M is the GMM Supervector, m is the UBM Supervector, T is the TV space, and x is the I Vector. The process is to randomly initialize a T matrix and compute x. Using a maximum likelihood estimate algorithm with the UBM, T can then be recalculated. This process is computationally demanding especially at large dimensions for T.

The function `train_tvspace(...)` is provided by the MSR Toolkit to estimate T. The dimension of T is user defined and a target for optimization. Once TV space is estimated, extracting the I Vector is simply calculating x. This is done in the MSR Identity Toolkit with the provided function `extract_ivector(...)`

3.2.3 Linear Discriminant Analysis

I Vectors can be dimensionally reduced further using Linear Discriminant Analysis (LDA). LDA projects data onto a lower-dimensional vector space such that the ratio of the between-class distance to the within-class distance is maximized. The dimension of LDA reduced data is $n - 1$, where n is the number of distinct classes in the data. Six emotion classes are used in this project, so the final sized feature vector is 5 dimensions in size obtained by reducing the I Vectors with the LDA function provided by the MSR Identity Toolkit.

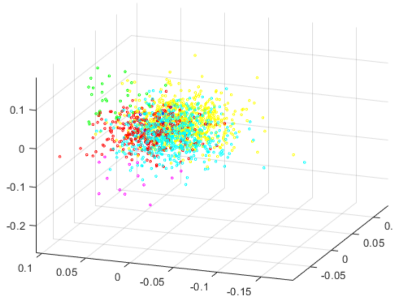

Figure 3.3: First 3 dimensions of 5D LDA reduced I Vectors

LDA has the advantage of visualizing data in its most linearly separable dimensions. Figure 3.3 presents a three dimensional plot of the LDA reduced I Vectors. Although there is some overlap between some classes, others, such as the green class, are in separate clusters. This is shown in more detail in the Results section of the paper.

3.3 Automatic Classification

3.3.1 Evaluation Metric

The evaluation metric used in the project is Unweighted Average Recall (UAR). For each utterance x_i belonging to emotion class C_n, x_i can either be classified correctly as C_n or incorrectly as another class C_m. These are referred to as true positive (TP) or false negative (FN), respectively. Recall (R) for class C_n is defined as

$$R_n = \frac{TP_{C_n}}{TP_{C_n} + FN_{C_n}} \tag{3.2}$$

and unweighted average recall for n classes is defined as

$$UAR = \frac{1}{n} \sum_{1}^{n} R_n \tag{3.3}$$

UAR is useful for its fast computability, equal weighting of imbalanced classes and commonality in other research often using it allowing comparisons to be made with said research. During its calculation per class accuracy is also calculated, so more insight is available on the classifier should it be needed.

Originally the weighted average recall (WAR) metric was considered as the primary evaluation metric. However it was prone to issues UAR was not. WAR is calculated similarly to UAR but requires class weight. The weight w of each class C_n is seen in Table 3.1, and were calculated as follows.

$$w_n = \frac{i_{C_n}}{i} \tag{3.4}$$

where i_{C_n} is the total number of labeled utterances for emotional class C_n and i is the total number of utterances in all classes. WAR is calculated as follows.

$$WAR = \sum_{1}^{n} w_n * R_n \tag{3.5}$$

WAR shares similar positives to UAR; fast computatability and common in experiments. WAR, however, prioritizes classes with a larger ratio in the data. When recognizing emotions, the objective is to recognize each emotion correctly regardless of rarity, so a step to handle the imbalance of data in the project was to favor UAR as an evaluation metric.

3.3.2 Baseline Classifiers

Nearest neighbor classification is used as the baseline classifier in this project. The first step to nearest neighbor is to find the mean vector μ_n of each emotional class C_n. For an n-dimensional vector, the mean vector will be n-dimensional. With test input data, a similarity measure sim_{ni} is calculated between a feature vector x_i and mean vectors μ_n. The mean vector with the lowest sim_{ni} output is deemed the nearest neighbor and the input is classified as such.

Two similarity measures are used; cosine similarity and euclidean distance. Cosine similarity is simply the angle between the feature vector x_i and mean vectors μ_n, calculated with the following equation.

$$sim_{ni} = \cos(\theta) = \frac{\vec{x_i} \cdot \vec{\mu_n}}{||\vec{x_i}||\ ||\vec{\mu_n}||} \tag{3.6}$$

Cosine similarity provides information on the direction of the feature and class; this may not be a relevant metric. In cases such as neutral and anger, they could portray similar features and only differ in intensity, so the direction would be similar but the distance will be notable. For this reason euclidean distance is also used a metric. Using both cosine similarity and euclidean distance can account for multiple situations.

Nearest neighbor classification does not need to be optimized each time the feature extraction parameters are changed, so it can act as a fast metric to compare relative accuracy changes in feature extraction parameters. It has poor performance in classification as it depends on the data already being linearly separable which will not always be the case. It was primarily used in this project to optimize feature extraction; system tests also found use with nearest neighbor due to its fast computation time.

3.3.3 Deep Neural Network

Four options were considered for the neural network; Deep NN, Convolutional NN, Recurrant NN or a hybrid. Due to I Vectors being an $n \times 1$ vector, a CNNs purpose of processing matrices was not necessary. Likewise, the sequential nature of the data is no longer present due to GMMs ignoring sequential data. RNNs aim to process such sequential data and would unnecessary in this project. With only one option remaning, a hybrid was not possible, so DNNs were selected.

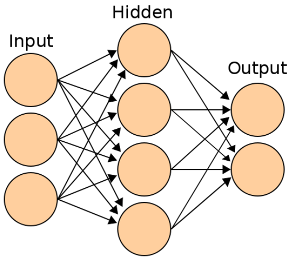

Figure 3.4: Example of a Neural Network

A DNN architecture is defined by its neurons, as in Figure 3.4. Each neuron in a layer uses the same activation function but has a distinct weight; this weight is the parameter learned during training. A DNN needs at least an input layer and an output layer. For this project, the output layer used is a softmax layer. For n classes, a softmax generates an $n \times 1$ vector with an entry corresponding to the likelihood an input belongs to a respective class. The highest likelihood is then taken as the label.

To implement and train the neural network, the Python Keras library was used. Keras offers easy to use graphics hardware acceleration which improves computation time allowing for more expensive optimizations to be trialed.

3.3.4 Optimization Process

Neural network optimization aims to improve overfitting or underfitting of data. A systematic approach was taken to optimization. Parameters of the network were defined, and a range of values was determined for each parameter. One at a time, each parameter would be changed in its range and the network would retrain while all other parameters were kept at their baseline value. The UAR differential with the baseline network is recorded and the experiment is repeated. The baseline values were selected based on the standards in research and a range slightly above and, if possible, below was used.

The parameters can be broken down into two categories; architecture based and learning based. Hidden layer count, feature count and dropout fall in the architecture based and the remainder in the learning based. The effects of both categories are observed and discussed in the results section.

Table 3.2: Parameter Ranges and Baselines

Parameter	Range	Baseline
Hidden Layer	1-4	1
Neuron Count	10-100	10
Epoch Count	10-100	30
Batch Size	10-125	30
Optimizer	SGD or Adam	SGD
Learning Rate	0.001-0.03	0.01
Feature Count	5 or 100	5
Dropout	0-0.5	0

3.3.5 Loss Function

The goal of a loss function is to provide a numerical value to represent the loss or cost of a classification algorithm during training. In this project, the classification made must be the correct emotion as close as possible, so a function that punishes mislabeling heavily is favored. The cross entropy (CE) function is one such function; it is designed to be high cost if the probability of the prediction y_p^k is not close to 1 for the index of its actual class value y_a^k. The function is defined as follows;

$$CE(y_p, y_a) = -\sum_{1}^{n} y_a^k \log y_p^k \tag{3.7}$$

where y_p is a vector containing the likelihood probability of each class is y_a is a zero vector containing a 1 in the index of its labeled class. Because of y_a only having one element that is not zero, the only value calculated by CE is the negative log of the likelihood of the predicted class being the correct class.

CE was problematic during testing because it suffered in performance with the imbalance of classes. More than 80% of the samples belong to one class, so more than 80% of the loss function calculations will be aiming to fit one class. This results in the classifier fitting all data into the one overpopulated class. A simple solution to this problem was using weighted cross entropy (WCE). Calculated similarly to cross entropy, weighted cross entropy takes into account the weight w of each class, giving more value to an individual prediction in a low sampled class versus an overpopulated class. This is calculated as follows;

$$CE(y_p, y_a, w) = -\sum_{1}^{n} w^k * y_a^k \log y_p^k \tag{3.8}$$

Results and Evaluation

All results shown are averaged from five trial runs with the aforementioned parameters. Table 4.1 displays results obtained from baseline and optimized experiments alongside the results from the original experiment. Neural networks perform similarly to LDA/Cosine Similarity as a discriminant and classifier algorithm. However, when using LDA/DNN, Neural Networks perform much better due to the reduced feature size minimizing overfitting. Although (Batliner et. al 2004) achieved similar impressive results with a large feature vector, the features were carefully selected and separable beforehand. This is not the case with the 100 dimension I Vector wherein the DNN overfits, but is the case for the dimensionally reduced variant as observed by the large differential in their accuracy.

As discussed earlier in the project, humans can disagree on labels. This can be observed by checking the label process of the data used. Each utterance has three labels. If only unanimous labels were used, there would only be a total of **2124** labels to use. That is only 40% of the data; therefore if humans can only completely agree on 40%, an automatic emotion recognition system can be expected to perform at least similarly.

Table 4.1: Results of Classifers with 80/20 train/test split

Classifier	Features	UAR
Cosine Similarity	5	24%
Euclidean Distance	5	46%
Neural Network (Batliner et. al 2004)	125	69%
DNN (Baseline)	100	24%
LDA/DNN (Baseline)	5	77%
LDA/DNN (Optimized)	5	81%

4.1 Maximum Linear Separability of Data

Figure 4.1 is the plot in all five dimensions, two dimensions at a time. Neutral, Angry and Emphatic utterances overlap in most dimensions, whereas the remainder are clearly separable in at least one dimensions. Neutral, Angry and Emphatic have the three largest samples out of the classes. This indicates the emotion of joyful, motherese or other is clearly expressed by the children despite the low number of times it is expressed, whereas the remainder have a chance to be confused with each other even by the labellers. The overlapping emotional classes could be closely matched but differ in a scale such as intensity.

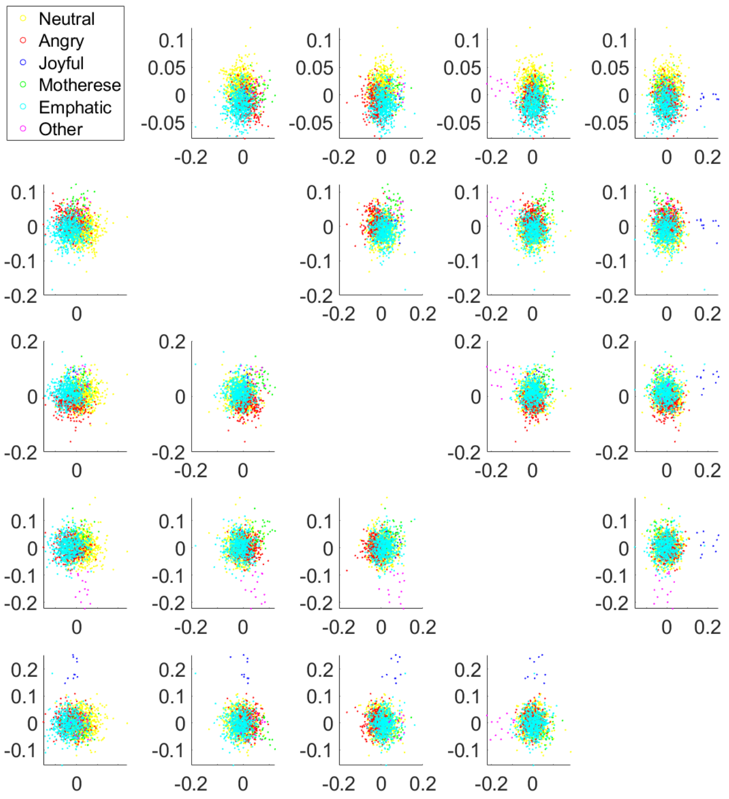

Figure 4.1: 2D plot of every dimension in an LDA reduced I Vector

4.2 Nearest Neighbor

4.2.1 Baseline Performance

Originally only cosine similarity was used as a baseline metric. However the data implied by Figure 4.1 could not be captured if the cluster means of the overlapping classes were too similar in direction. More information on these classes can be obtained using the distance to their means rather than direction, so euclidean distance was used as another similarity measure. The increase in performance with distance conveys there is potential in experimenting with an intensity based emotion classifier rather than a distinct equivalence class based emotion classifier as distance most likely relates to similar emotions with varying intensities.

4.2.2 Optimizations

4.2.3 Feature Extraction Optimizations

Table 4.2 summarizes parameters tweaked during feature extraction and changes to accuracy using cosine similarity. Shorter frame lengths, usually 20ms to 30ms, are favored in speech processing, yet interestingly performance improved with a longer than usual frame length. Since

Table 4.2: Feature Pramaters and UAR of Cosine Similarity

Frame Length (ms)	30	20	40	30	30	30	30	**40**
UBM Components	64	64	64	128	512	64	64	**512**
I Vector Dimensions	100	100	100	100	100	50	200	**100**
Cosine Similarity	21%	21%	23%	21%	22%	20%	21%	**24%**

MFCCs were originally designed for speech recognition tasks, this could indicate emotion data is expressed on a longer frame of speech than language. This could also be the case for children's speech and not only emotional data. The UBM and I Vector parameters are standard. Although using 200 dimensions for the I Vector improved performance, the computation time was drastically higher and unreasonable to test over a longer period of time or with a larger dataset, so the final dimensions used during DNN optimization was 100 dimensions.

4.3 Neural Network

4.3.1 Baseline Performance

Table 4.3: Perclass basline accuracy using weighted cross entropy loss function

Class	Angry	Joyful	Motherese	Emphatic	Neutral	Other
Accuracy	73.6%	100%	100%	56.6%	52.7%	100%

4.3.2 Loss Function

When originally using cross entropy as the loss function, baseline performance for LDA/DNN was approximately 20%, just above one-sixth, or the ratio of one class to all classes. This was definitely due to the class imbalance negatively impacting cross entropy. Once weighted cross entropy was implemented, the UAR improved drastically, avereging at 77%. While this is close to 5/6, or the ratio of all but one class, the class imbalance was not the factor as the per-class accuracy was more balanced between the classes. The per-class accuracy is listed in Table 4.3 alongside weights to clarify the effect of weighted cross entropy.

Table 4.4: Per-parameter Optimization

Parameter	Value	UAR	Baseline Differential
Basline	-	77%	-
Hidden Layers	2	73%	-4%
	3	63%	-14%
	4	42%	-25%
Layer Size	***20***	78%	+1%
	30	72%	-5%
	50	75%	-2%
	80	76%	-1%
	100	70%	-7%
Epoch Count	***10***	79%	+2%
	50	75%	-2%
	75	74%	-3%
	100	74%	-3%
Batch Size	10	44%	-33%
	50	77%	-
	75	76%	-1%
	100	76%	-1%
	125	74%	-3%
SGD + LR	0.001	73%	-4%
	0.003	77%	-
	0.006	78%	+1%
	0.009	77%	-
	0.03	42%	-35%
Adam + LR	0.001	73%	-4%
	0.003	77%	-
	0.009	78%	+1%
	0.01	77%	-
	0.03	54%	-23%
Dropout	0.1	76%	-1%
	0.2	78%	+1%

4.3.3 Parameter Optimization

Epoch count, batch size and learning rate displayed alternate behaviours when layer size, dropout and hidden layer count were at optimized rather than baseline levels. The values with the best differential for the learning based parameters before any optimizations were 30, 10, and 0.001 for Epoch count, batch size and learning rate, respectively. The lower learning rate and increased iterations accounted for the higher layer size and lower dropout at baseline by learning slower but over a longer period of time. Since the latter are architecture based parameters that affect the base architecture of the DNN, changes made to them can propagate to the learning step. Results show it was best to optimize architecture based parameters for this reason. The final most optimal combination is presented in Table 4.5

Table 4.5: Most optimal DNN parameters found

Parameter	Value
UAR	82%
Hidden Layers	1
Layer Size	20
Epochs	10
Batch Size	50
Learning Rate	SGD - 0.006
Feature Size	LDA - 5 dim
Dropout	0.2

Conclusion

5.1 Reviewing Objectives

The primary objective of this project was to build a system to automatically recognize emotions expressed in children's speech using acoustic speech data using the latest state of the art techniques of I Vectors and neural networks. This project achieved this by:

1. Providing a review of current techniques and the reasons they are used over older techniques.
2. Breaking down the data used substantially to allow consideration for any abnormal situations or labels.
3. Used modern I Vector and neural networks techniques and provided evidence to the improvement in accuracy they provide over other techniques.
4. Evaluated results and proposed areas to continue from this project, such as using a scale definition of emotion over a discrete definition.

5.2 Findings and Contributions

Emotion recognition is not at the level of speech recognition in popularity, usage and research, but with recent advancements it is improving rapidly. Some findings in this project could contribute to the field of emotion recognition.

A larger than normal MFCC frame size of 40ms exhibited improved performance, which can be considered unusual. Most speech processing techniques deal with either recognition or identification/verification, so experimenting with alternate framze sizes or cepstral coefficients can present new state of the art emotion recognition techniques.

Neural Network performance is more effective with LDA reduced features of a small size than hand picked features of a large size. Using an alternate discriminator to process features before classifying with a neural network is not limited to emotion recognition; this behavior can be true for various machine learning problems.

5.3 Further Steps

Alternative approaches can be taken to accommodate for class imbalance. Data augmentation using frequency normalization to increase the sample size of classes besides Neutral could prove effective. Alternative neural network architecture, such as hybrid builds, will provide more results to use in understanding effective classification techniques. The dataset provides both English and German speech data of the same type, so using the methods from this project to see results on cross-lingual training and predicting can advance on the issue of cross lingual and cultural emotion recognition.

Bibliography

Alex, S.B., Babu, B.P., Mary, L., 2018. Utterance and Syllable Level Prosodic Features for Automatic Emotion Recognition, in: 2018 IEEE Recent Advances in Intelligent Computational Systems (RAICS). Presented at the 2018 IEEE Recent Advances in Intelligent Computational Systems (RAICS), pp. 31–35. https://doi.org/10.1109/RAICS.2018.8635059

Alim, S.A., Rashid, N.K.A., 2018. Some Commonly Used Speech Feature Extraction Algorithms. From Natural to Artificial Intelligence - Algorithms and Applications. https://doi.org/10.5772/intechopen.80419

Al-Kaltakchi, M.T.S., Woo, W.L., Dlay, S.S., Chambers, J.A., 2017. Comparison of I-vector and GMM-UBM approaches to speaker identification with TIMIT and NIST 2008 databases in challenging environments, in: 2017 25th European Signal Processing Conference (EUSIPCO). Presented at the 2017 25th European Signal Processing Conference (EUSIPCO), pp. 533–537. https://doi.org/10.23919/EUSIPCO.2017.8081264

Batliner, A., Blomberg, M., D'Arcy, S., Elenius, D., Giuliani, D., Gerosa, M., Hacker, C., Russell, M., Steidl, S., Wong, M., 2005. The PF_STAR children's speech corpus. pp. 2761–2764.

Batliner, A., Hacker, C., Steidl, S., Nöth, E., D'Arcy, S., Russell, M., Wong, M., 2004. "You Stupid Tin Box" - Children Interacting with the AIBO Robot: A Cross-linguistic Emotional Speech Corpus, in: Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04). Presented at the LREC 2004, European Language Resources Association (ELRA), Lisbon, Portugal.

Batliner, A., Schuller, B., Seppi, D., Steidl, S., Devillers, L., Vidrascu, L., Vogt, T., Aharonson, V., Amir, N., 2011. The Automatic Recognition of Emotions in Speech, in: Cognitive Technologies. pp. 71–99. https://doi.org/10.1007/978-3-642-15184-2_6

Cabanac, M., 2002. What is emotion? Behavioural Processes 60, 69–83. https://doi.org/10.1016/S0376-6357(02)00078-5

Casale, S., Russo, A., Scebba, G., Serrano, S., 2008. Speech Emotion Classification Using Machine Learning Algorithms, in: 2008 IEEE International Conference on Semantic Computing. Presented at the 2008 IEEE International Conference on Semantic Computing, pp. 158–165. https://doi.org/10.1109/ICSC.2008.43

Chen, C.-Y., Chen, C.-P., 2016. Support Super-Vector Machines in Automatic Speech Emotion Recognition 11.

Cowen, A.S., Keltner, D., 2017. Self-report captures 27 distinct categories of emotion bridged by continuous gradients. PNAS. https://doi.org/10.1073/pnas.1702247114

Frick, R.W., 1985. Communicating emotion: The role of prosodic features. Psychological Bulletin 97, 412–429. https://doi.org/10.1037/0033-2909.97.3.412

Gao, M., Dong, J., Zhou, D., Zhang, Q., Yang, D., 2019. End-to-End Speech Emotion Recognition Based on One-Dimensional Convolutional Neural Network, in: Proceedings of the 2019 3rd International Conference on Innovation in Artificial Intelligence, ICIAI 2019. Association for Computing Machinery, Suzhou, China, pp. 78–82. https://doi.org/10.1145/3319921.3319963

Gupta, S., 2016. Application of MFCC in Text Independent Speaker Recognition.

Han, K., Yu, D., Tashev, I., 2014. Speech Emotion Recognition Using Deep Neural Network and Extreme Learning Machine 5.

Heracleous, P., Yoneyama, A., 2019. A comprehensive study on bilingual and multilingual speech emotion recognition using a two-pass classification scheme. PLoS ONE 14, e0220386.
https://doi.org/10.1371/journal.pone.0220386

Hu, H., Xu, M.-X., Wu, W., 2007. GMM Supervector Based SVM with Spectral Features for Speech Emotion Recognition, in: 2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07. Presented at the 2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07, pp. IV-413-IV–416. https://doi.org/10.1109/ICASSP.2007.366937

Ibrahim, N.S., Ramli, D.A., 2018. I-vector Extraction for Speaker Recognition Based on Dimensionality Reduction. Procedia Computer Science, Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 22nd International Conference, KES-2018, Belgrade, Serbia 126, 1534–1540.
https://doi.org/10.1016/j.procs.2018.08.126

Issa, D., Fatih Demirci, M., Yazici, A., 2020. Speech emotion recognition with deep convolutional neural networks. Biomedical Signal Processing and Control 59, 101894.
https://doi.org/10.1016/j.bspc.2020.101894

Jin, Q., Li, C., Chen, S., Wu, H., 2015. Speech emotion recognition with acoustic and lexical features, in: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Presented at the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4749–4753. https://doi.org/10.1109/ICASSP.2015.7178872

Karafiát, M., Burget, L., Matejka, P., Glembek, O., Cernocky, J., 2011. iVector-based discriminative adaptation for automatic speech recognition. https://doi.org/10.1109/ASRU.2011.6163922

Kinnunen, T., Li, H., 2010. An Overview of Text-Independent Speaker Recognition: from Features to Supervectors. Speech Communication 52, 12–40. https://doi.org/10.1016/j.specom.2009.08.009

Kołakowska, A., Landowska, A., Szwoch, M., Szwoch, W., Wróbel, M., 2014. Emotion Recognition and Its Applications. Advances in Intelligent Systems and Computing 300, 51–62. https://doi.org/10.1007/978-3-319-08491-6_5

Lim, W., Jang, D., Lee, T., 2016. Speech emotion recognition using convolutional and Recurrent Neural Networks, in: 2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA). Presented at the 2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA), pp. 1–4.
https://doi.org/10.1109/APSIPA.2016.7820699

Markel, J., Oshika, B., Gray, A., 1977. Long-term feature averaging for speaker recognition. IEEE Transactions on Acoustics, Speech, and Signal Processing 25, 330–337.
https://doi.org/10.1109/TASSP.1977.1162961

Mehta, D., Siddiqui, M.F.H., Javaid, A.Y., 2019. Recognition of Emotion Intensities Using Machine Learning Algorithms: A Comparative Study. Sensors (Basel) 19. https://doi.org/10.3390/s19081897

Mesquita, B., Walker, R., 2003. Cultural differences in emotions: a context for interpreting emotional experiences. Behaviour Research and Therapy 41, 777–793. https://doi.org/10.1016/S0005-7967(02)00189-4

Pérez-Espinosa, H., Martínez-Miranda, J., Espinosa-Curiel, I., Rodríguez-Jacobo, J., Villaseñor-Pineda, L., Avila-George, H., 2020. IESC-Child: An Interactive Emotional Children's Speech Corpus. Computer Speech & Language 59, 55–74. https://doi.org/10.1016/j.csl.2019.06.006

Plutchik, R., 1982. A psychoevolutionary theory of emotions: Social Science Information.
https://doi.org/10.1177/053901882021004003

Rajan, R., G., H.U., C., S.A., M., R.T., 2019. Design and Development of a Multi-Lingual Speech Corpora (TaMaR-EmoDB) for Emotion Analysis, in: Interspeech 2019. Presented at the Interspeech 2019, ISCA, pp. 3267–3271. https://doi.org/10.21437/Interspeech.2019-2034

Reddy, A.P., Vijayarajan, V., 2020. Audio compression with multi-algorithm fusion and its impact in speech emotion recognition. Int J Speech Technol. https://doi.org/10.1007/s10772-020-09689-9

Rizzo, A., Morie, J.F., Williams, J., Pair, J., Buckwalter, J.G., 2005. Human Emotional State and its Relevance for Military VR Training. https://doi.org/null

Russell, M., 2006. The PF-STAR British English Children's Speech Corpus 35.

Sabini, J., Silver, M., 2005. Ekman's basic emotions: Why not love and jealousy? Cognition & Emotion 19, 693–712. https://doi.org/10.1080/02699930441000481

Sadjadi, S.O., Slaney, M., Heck, L., 2013. MSR Identity Toolbox.

Schwenker, F., Scherer, S., Magdi, Y.M., Palm, G., 2009. The GMM-SVM Supervector Approach for the Recognition of the Emotional Status from Speech, in: Alippi, C., Polycarpou, M., Panayiotou, C., Ellinas, G. (Eds.), Artificial Neural Networks – ICANN 2009, Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 894–903. https://doi.org/10.1007/978-3-642-04274-4_92

Wierzbicka, A., 1992. Defining emotion concepts. Cognitive Science 16, 539–581. https://doi.org/10.1016/0364-0213(92)90031-O

Xia, R., Liu, Y., 2016. DBN-ivector Framework for Acoustic Emotion Recognition. Presented at the Interspeech 2016, pp. 480–484. https://doi.org/10.21437/Interspeech.2016-488