Justin Ross

0050040865

CSC8003 – Machine Learning

Assignment 1

Contents

Introduction

Background

Determining emotional sentiment from audio has been a longstanding project for computer scientists. Although the topic has long been of interest the techniques available before machine learning were of limited effectiveness[1].

Significance

There are a number of great potential applications from improving business sales to identifying people that may be suffering from depression[2]. The technology can even be used to detect the emotional sentiment of music and therefore suggest music content based on the emotional state of the listener[3].

My Choice

I chose this project as I believe that technology products are effective when they successfully tap into emotions. Companies like Instagram and Facebook are successful because they cater to people's emotional desires as people largely make decisions using emotions.

Problem Solution

In this project we are prohibited from using deep learning techniques, so will attempt to solve the problem using classical machine learning techniques. My planned approach is to use a dummy classifier to set a benchmark and then to apply a series of different approaches before picking the one with the highest accuracy, mostly likely a random forest.

Literature Review

Feature Selection

Feature selection and engineering are the most important parts of speech emotion recognition using classical techniques. Two approaches are dominant: creating a feature vector and using a spectrogram[4].

Creating a feature vector involves dividing the speech signal into segments, a process called 'framing'[5]. After which the idea is to create a vector using aspects of the frame that enable the emotional

[1] (Hajarolasvadi and Demirel 2019, p.1)
[2] (Crangle et al., 2019, p. 2)
[3] (Hajarolasvadi and Demirel 2019, p.1)
[4] (Hajarolasvadi and Demirel 2019, p. 10)
[5] (Wang 2020, p.2)

sentiment of the user to be identified. For example volume could be one component of the vector and may help to identify anger. This approach came from the pre-machine learning era of computing.

The spectrogram approach involves generating a spectrogram, which is a 2d visual representation of a short-time Fourier transform (STFT) with the x-axis representing time and the y-axis representing signal frequency[6]. The advantage of the log mel scale spectrogram is that the mel spectrum it is designed around the spectrum of sound that humans hear as being equally distant apart and the log transform emphasizes the lower frequency sounds that humans place more emphasis on[7]. As such it captures the data important for humans to differentiate sounds and therefore emotions.

Model Selection

In all of the literature cases deep learning techniques such as Convolutional Neural Networks (CNN) were eventually utilized to solve the problem rather than classical machine learning techniques due to the low accuracy of classical in this particular case[8]. However, we need to apply a classical method and a study from Stanford University's Center for the Study of Language and Information suggested that a random forest is a good choice[9]. When it came to building the model the Stanford team used a sophisticated approach that involved breaking down the audio data into components that reflect emotion: jittery-ness of the voice, shimmer and periodicity[10]. For this project I will not go into as much depth in feature engineering as the Stanford team.

Problem Statement

The input data is a set of voice recordings in .wav format that have been converted into an array of log mel spectrogram averages and from those average values the objective is to output which of the eight emotional categories they belong to with as much accuracy as possible. Emotional category outputs include: neutral, calm, happy, sad, angry, fearful, disgusted and surprise.

Dataset

Class distribution is missing. How many data are available for each of the classes?

Details

The data is in .wav format and is from the RAVDESS project[11], which is a data set of 24 actors, 12 male and 12 female, repeating the same line with one of eight different emotional classifications: neutral, calm, happy, sad, angry, fearful, disgusted and surprised. Once transformed to a dataframe, the dataset has 1440 rows and 262 columns.

---

[6] (Hajarolasvadi and Demirel 2019, p. 2)
[7] (Choi, Fazekas, Cho and Sandler, 2017, p.1).
[8] (Hajarolasvadi and Demirel 2019, p.2)
[9] (Crangle et al., 2019, p. 3)
[10] (Crangle et al., 2019, p. 3)
[11] https://smartlaboratory.org/ravdess/

There are further class distinctions with all but the non-neutral data having a split between normal and strong intensity and odd numbered voice actors are male and even numbered are female. Overall, though, we are mostly interested in differentiating between the emotional classes.

Visualization/ Exploratory Data Analysis (EDA)

	gender	emotion	actor	0	1	2	3	4	5	6	...	249	250	251	252
0	male	neutral	1	-76.384773	-76.384773	-76.384773	-76.384773	-76.384773	-76.384773	-76.384773	...	0.000000	0.000000	0.000000	0.000000
1	male	neutral	1	-75.335518	-75.445320	-75.554031	-75.203949	-75.230530	-75.319374	-75.653793	...	0.000000	0.000000	0.000000	0.000000
2	male	neutral	1	-75.150711	-75.150711	-75.150711	-75.150711	-75.150711	-75.150711	-75.150711	...	0.000000	0.000000	0.000000	0.000000
3	male	neutral	1	-75.268448	-75.268448	-75.268448	-75.268448	-75.268448	-75.268448	-75.268448	...	0.000000	0.000000	0.000000	0.000000
4	male	calm	1	-80.147377	-80.147377	-80.147377	-80.147377	-80.147377	-80.147377	-80.147377	...	-80.147377	-80.147377	-80.121956	-79.998009

Figure 1: First five rows of the dataframe. Note it is contracted due to number of columns.

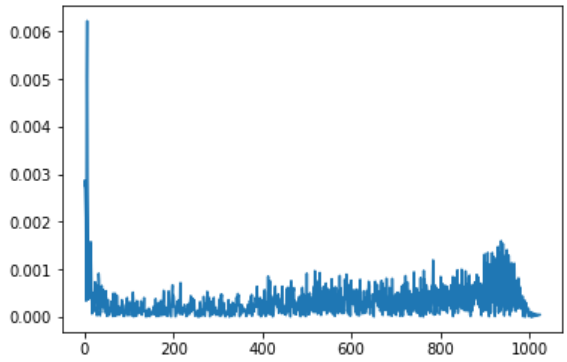

Figure 2: Fourier Series representation of voice recording.

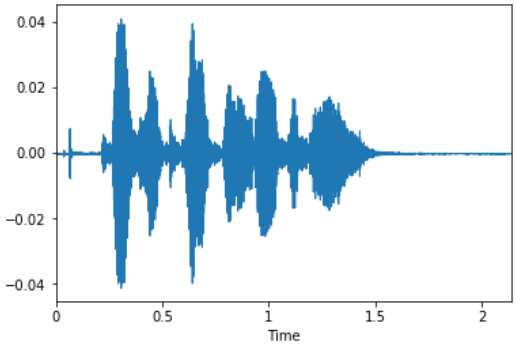

Figure 3: Log mel scale spectrogram of a voice recording

Benchmark

Benchmark Selection

The benchmark is the accuracy score for scikitlearn's dummy classifier. The stratified dummy classifier makes predictions on the basis of guessing based on the class distributions of the data. In other words, as the data contains 8 classes of emotions we are expecting the dummy classifier to be accurate roughly $1/8^{th}$ of the time. Secondary benchmarks are the precision, recall and f1 score for the dummy classifier.

Dummy classifier is very simple. You should always choose a ml model.

Benchmark Justification

The value in using the dummy classifier is that it gives us the chance of making an accurate prediction if we simply calculate the ratio of the classes and guess. It is therefore the perfect starting point.

Evaluation Metric

Selection

The primarily evaluation metric will be accuracy as it is a good metric for this particular problem because we are trying to determine to which emotional category the speech belongs. Accuracy is a good metric

You should mention the problem of accuracy more clearly with examples

for this particular case as it is primarily designed to measure whether the set of predicted labels match the real labels. In other words, it is designed to test whether the algorithm developed using the training data was able to predict correct labels for the test data. It is calculated with the following formula:

$$Accuracy = \frac{True\ Negatives + True\ Positive}{True\ Positive + False\ Positive + True\ Negative + False\ Negative}$$

Other metrics that could be useful were also calculated, such as F1Score, Recall and Precision:

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$F1 - Score = \left(\frac{Recall^{-1} + Precision^{-1}}{2}\right)^{-1} = 2 \cdot \frac{(Precision \cdot Recall)}{(Precision + Recall)}$$

Justification

Part of the reason that we are able to go with accuracy is that we are more interested in true positives and true negatives than in false positives and false negatives, but this could change depending on what we want to use this data for. For example, if we were using this data to try to identify the emotional state of people calling a psychologist or helpline of some kind then false negatives could be particularly harmful. As such, in that case we may prefer the f1 score as it is a superior metric that focuses more on false positives and negatives. As a result I have used both of these metrics, but lean more toward accuracy.

Methodology

Data Preparation

The first step is that we needed to load the data into a dataframe, taking into account the file name to label the data appropriately. Our first step is to iterate through all the files, separated as they are by different folders for different actors. Then we need to use the file name itself to determine the various labels the data has.

The second step after creating the iteration loop was to use the file name to label the data. The file names followed a specific format, eg '03-01-01-01-01-01-01.wav'. Then we need to divide the actor number by 2 to determine whether it is an odd or even number, with odd numbered actors given the gender identity of male and even numbered actors given the gender identity of female. After which the various numbers in the file name determine the actor number, gender and emotion being expressed.

The third step was to transform the audio data itself into something workable. There were a few options but ultimately it was decided to use a log mel spectrogram transformation on the data using the librosa package. The reason for this is we need a way to represent the data numerically. To improve efficiency in this process only 3 seconds of the audio was used starting at the 0.5 second mark and utilizing a sampling rate of 44100 hz.

The fourth step occurred after converting the data to the log mel spectrogram and involved generating a series of mean values which would become the columns. Although there is information loss using this process it is still required to reduce the sheer volume of data that the audio file contains. The end result is a series of mean values that, combined, represent the audio file and can be used to predict the category to which it belongs.

Feature Selection

As per the literature, feature selection is most important and also the hardest part of this particular project. The problem is that it does not have clear features that might allow categorization. It is not like housing prices where number of bedrooms would be a strong predictor of price. The features themselves are quite vague though this vagary also makes it suitable for machine learning. In our case the data preparation steps largely took care of feature selection, the most complicated component of which was generating the mean values for the partitioned log mel spectrogram.

Awesome! I just love it.

Class Imbalance

There is a very slight, potential class imbalance issue as the neutral emotion data does not differentiate between strong and normal intensity. This is not a real problem though as neutrality is inherently not strong in intensity, the neutral emotion set to strong or normal intensity can be seen to be the same. Overall the data is balanced and well distributed.

Algorithms

You are suppose to use at least two algorithms so the discussion should include two algorithms :-(

Decision Tree Ensemble (Random Forest) Algorithm

Scikitlearn's decision tree utilizes a modified version of the Classification and Regression Tree (CART) algorithm with the Gini index used as its default metric as the splitting criteria for nodes. The calculation formula is as follows:

$$Gini(D) = 1 - \sum_{i=1}^{c} (p_i)^2$$

Where p is the relative frequency of class i in the data, D is the data and c is the number of classes.

The objective of using the Gini index is to guide how the branches of the decision tree split by generating an index value for each feature based on how many correct predictions of the class it makes and then selecting for the lowest value. If one feature split, for example on some specific mean value for the log mel spectrogram, resulted in a perfect classification of emotion then it would in principle have a Gini index of 0. Hence the selection of the lowest Gini index is otherwise referred to as impurity reduction.

The random forest uses a collection of decision trees created with different subsets of the training data to create 'bags' of values and then averages the values in the bag to create the model. This expressed with the following formula in regression cases:

$$\hat{f} = \frac{1}{B}\sum_{b=1}^{B} f_b\,(x')$$

However, in cases such as ours this is done by creating a bag of values and choosing the most common. Thus, if 3/6 of the trees selected anger as the emotion and 2/6 trees suggested neutral and 1/6 suggested fear then anger would be the selection as it is the most common. It functions much like a vote based on the different trees predictions based on specific aspects of the independent variable.

Advantages

- Has many of the strength of decision trees, but lowers the negatives through randomization.
- Low risk of overfitting because each tree is based on a random sub-sample of the training data.
- The random forest is robust to outliers because every variation of the decision tree differs.

Disadvantages

- The random forest can falter when one feature of the data is of extreme importance compared to the others because the algorithm chooses features to ignore at each node.
- Does not deal well with general changes over time otherwise called.
- Can struggle when there are a vast number of different categories.

Justification

Before justifying the algorithm it is important to remember that there are many different algorithms, with many different parameters. As such it is difficult to truly say that this was the best algorithm, but its performance was quite good in comparison to the others tested and it was the choice of a group of Stanford researchers so seems like a fair choice[12].

[12] (Crangle et al., 2019, p. 3)

Training and Hyperparameter Tuning

Hyperparameters

For the random forest there are three different hyperparameters that are important: the depth of the decision tree, the number of trees used and the number of variables chosen randomly. The intuition behind the depth of the decision tree is that it corresponds to how many questions the decision tree poses of the data in order to guess its category. The number of trees is how many decision trees are used in the ensemble process of the random forest, with more trees generally being better. The number of variables chosen determines how many variables are used at each splitting question.

Training and Test Sets

The training and test sets were split 80/20, stratified by the emotion, actor and gender categories. There is probably room to optimize based on trying some different training/test ratios, but 80/20 seems good as this data set does not have so much data. The conventional split is something closer to 70/30, but in this case there is not much data and therefore 80% of the data will be used for training. Concerning stratification the aim is to preserve the pre-existing balance of the data. If an 80/20 split was chosen randomly it may become unbalanced and it also would not be appropriately blocked, risking overfitting.

Tuning Hyperparameters

A grid search, otherwise called a parameter sweep, was used to tune the hyperparameters. The grid search being an iteration through different values for the hyperparameters until the combination with the highest accuracy was given. A tree depth ranging from 1 to 25 in steps of 1 was tested, number of trees ranging from 50-500 in steps of 50 were tested and number of variables ranging from 10 to 50 in steps of 10 were tested. This meant going through 1,250 different variations until the optimal set of hyperparameters was chosen. Computing and time constraints prevent further optimization.

You have provided a good analysis. But you need to use two algorithms and get final results for both of the models after hyperparameter tunning. After that you compare and select the best one. Then compare that one with the benchmark.

Results

	Final Model	Initial Model	Benchmark
Accuracy	0.5173611111111112	0.4652777777777778	0.0763888888888889
F1 Score	0.5120132682982861	0.4626830083730994	0.12332486386143894
Recall	0.5168827246129878	0.46356684004710325	0.12179029629687524
Precision	0.5144724892517748	0.46670488438062124	0.1257292745478721

Table 1: Metrics compared for benchmark, initial and final model. Higher is better.

Initial Model Performance

The initial model performance was based on scikitlearn's default random forest, which performed with an accuracy of roughly 46.52% (rounded down).

Best Performing Hyperparameter (Final Model)

The best performing hyperparameters were a depth of decision tree of 16, number of trees 250 and number of variables chosen randomly 20.

Final Model Performance

The final model offered an accuracy performance improvement of 5.2% (rounded down) over the initial model. Similar improvement occurred for secondary metrics such as F1 Score, Recall, etc as per table 1.

Models vs Benchmark

The random forest models significantly outperformed the accuracy of the benchmark, offering performance improvements of 38.88% in the initial model's case and 44.09% in the final model's case, both figures rounded down.

Conclusion

Project Summary

The project proved to be more successful than anticipated with the final model having an accuracy of roughly 52%. Although this may not seem very accurate it is worth noting that there are 8 emotional categories and the benchmark, which makes a guess based on chance, achieved a lowly 7.6% accuracy.

Challenges

The biggest challenge that I faced doing this project was making decisions about what not to use or not to include. The sheer number of options for algorithms, hyperparameter tuning, test/training split, stratification and for loading the data was staggering. Often, there is not really a clear sign that one method or process will be superior to the others and there is a large element of trial-and-error to the entire process. This is further exacerbated by delays due to waiting on the computer to complete the work, which might have been solved using a random search rather than a grid search. The hyperparameter tuning aspect was very heavily affected by lack of computing power. I have little doubt that improvement could be made with more iterations, but at the cost of efficiency.

References

Choi, K., Fazekas, G., Cho, K. and Sandler, M., 2017. A Comparison of Audio Signal Preprocessing Methods for Deep Neural Networks on Music Tagging. arXiv Pre-print,.

Crangle, C., Wang, R., Perreau-Guimaraes, M., Nguyen, M., Nguyen, D. and Suppes, P., 2019. Machine learning for the recognition of emotion in the speech of couples in psychotherapy using the Stanford Suppes Brain Lab Psychotherapy Dataset. arXiv Pre-print,.

Hajarolasvadi, N. and Demirel, H., 2019. 3D CNN-Based Speech Emotion Recognition Using K-Means Clustering and Spectrograms. Entropy, 21(5), p.479.

Wang, C., 2020. Speech Emotion Recognition Based on Multi-feature and Multi-lingual Fusion. arXiv Pre-print,.