UNIVERSITY OF CALIFORNIA, LOS ANGELES

Cry, Cry, Cry

Kyle Colton
Christian Gao
Benjamin Hong
Feiran Zhu

Contents

1 Abstract 3

2 Questions 3

3 Variable Description 3

4 Statistical Methods Used 3

4.1 Hidden Markov Model 3

5 Summary of Findings 4

6 Classification using Pitch Analysis 4

6.1 K-Means Applied to Pitch Estimates 5

6.1.1 Performance of K-Means 5

6.2 Support Vector Machine Applied to Pitch Estimates 5

6.2.1 Performance of SVM 6

6.3 Using Baum-Welch Algorithm 6

7 Conclusion 6

8 Shortcomings 6

9 Recommendations 6

Objective. The purpose of this study is to discuss algorithms designed to predict whether or not a baby is crying.

Data Collection and Procedure. We were given a multitude of audio files (in WAV format) of animal, baby, and adult sounds. We analyzed the pitch of each sound using two different procedures: (1) using FOSS *aubio* and conducting K-means clustering, which is an iterative classification method that we will discuss in more detail later, and support vector machine (SVM) classification and (2) using a specially designed pitch extraction algorithm to ultimately use the Baum-Welch algorithm to answer a number of questions that we will make clear throughout the report.

Conclusion. We arrived at a decent classification algorithm using K-means and SVM, but because of limitations in the data, we did not end up with an extremely successful algorithm based on HMM.

1 Abstract

Caring for a baby requires patience and effective communication. A caretaker attends to the baby when he/she cries, but this strictly involves an auditory process. We can see how this becomes problematic for parents who are deaf or have difficulty hearing. Fortunately, because of technological advances, there are now devices that assist parents in determining whether or not their child is crying. Of course, for a device to accurately decipher noises, the algorithm must be able to distinguish between different noises (e.g., between a dog crying and a baby crying). This report discusses the algorithms we designed to predict the baby's emotional state—that is, is the baby crying, laughing, or neutral?

2 Questions

1. Can we design an algorithm that accurately predicts whether or not a baby is crying?
2. Given that our algorithm works, what is its prediction rate—that is, how often does our algorithm successfully classify noises into their respective categories?

3 Variable Description

4 Statistical Methods Used

4.1 Hidden Markov Model

	Cry	Neutral	Laugh
Cry	0.8	0.15	0.05
Neutral	0.2	0.6	0.2
Laugh	0.05	0.35	0.6

↓

	Cry	Neutral	Laugh
Cry	0.85	0.1	0.05
Neutral	0.15	0.7	0.15
Laugh	0.1	0.4	0.5

Let X_t be a discrete hidden random variable with N possible values. $P(X_t|X_{t-1})$ is independent of time t, so our transition matrix will be:

$$A = \{a_{ij}\} = P(X_t = j|X_{t-1} = i) \tag{4.1}$$

The initial state distribution at, for example, $t = 1$, is given by: $\boxed{\pi_i = P(X_1 = i)}$

The probability of a certain observation occurring at time t for state j is given by: $\boxed{b_j(y_t) = P(Y_t = y_t|X_t = j)}$

The observation sequence should look like $Y = (Y_1 = y_1, Y_2 = y_2, \ldots, Y_t = y_t)$

Altogether, we now have a hidden Markov chain that can be described by:

$$\boxed{\theta = (A, B, \pi)} \tag{4.2}$$

5 Summary of Findings

6 Classification using Pitch Analysis

Pitch detection and analysis was explored as a method to identify baby cries. The detection method relies on the assumption that pitch patterns for each sound type will differ, but within the sound types each sample will be similar.

Raw audio files (stored as `wav`) were run through a Free/Libre and Open Source Software package called Aubio[1]. Aubio is a C-based audio labeling tool that includes several methods and controllable methods for pitch detection in an audio stream.

In order to complete pitch analysis, each 1.5 second `wav` file was fed through Aubio to get time-series pitch data, then broken into “events” based on silence. In figure 6.1, we see an estimate of pitches over time for one such file. This file, for example would be broken into two events. The main event stretches from approximately $t = 0.00$ to $t = 0.85$. A second event occurs around $t = 1.40$.

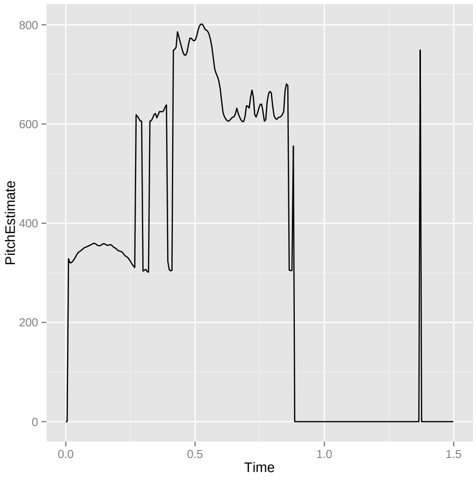

Figure 6.1: Pitch estimate of a `wav` file

Looking at the graph, it appears that the second event in this file is an error. It is a blip caused by something in the recording that we can safely ignore if we can identify it and separate it from the good data. An easy approximation is to ignore all events shorter

[1] Package information can be found at `aubio.org` or at http://git.aubio.org

than a set time interval. After testing all time intervals using a given model, $t = 0.1$ seconds was chosen as the cutoff for "real" versus "fake" events.

```
> head(baby_laugh01,5)
        V1       V2
1 0.000000   0.0000
2 0.005333   0.0000
3 0.010667 328.1827
4 0.016000 320.3024
5 0.021333 320.4278
```

Figure 6.2: Data from Aubio with time (l) and pitch (r)

After events were detected and pre-processed, summary statistics were collected on each event. Each event was stored with filename, event type (baby_cry, baby_laugh, etc.), minimum pitch, 1st quartile, median pitch, 3rd quartile, maximum pitch, mean pitch, and length of the event. Models were then run on these summary statistics to capture and classify them and compare them to the originating source. To start with, the models were built around using the events individually. Eventually, smarter models were built to consider all events from a single file as a unit and decide by voting on the predicted classification of those files.

A baby detector, by nature, is attempting to find when the baby is unhappy by listening for crying. With this in mind, the power of the model (the correct identification of an event when one occurs) is far more important than any other statistics[2]. For this reason, the models assume that if one event is categorized as a `baby_cry` event, then that file is a `baby_cry` event. This increases power at the expense of accuracy.

Additionally, the model was simplified into a binary outcome of a baby crying or not crying. While model was able to classify with some success the other events, the purpose of the monitor is to detect a baby crying, so the model was simplified.

```
> events[1:3,2:9]
     Sound.Source   Min    X1Q Median Mean  X3Q  Max Length
1 Adultfemale_cry 149.3  504.3   1848 1419 1897 3596     77
2 Adultfemale_cry 400.6 1110.0   1885 1913 2906 4165     30
3 Adultfemale_cry 149.3  553.3   1840 1424 1898 3602    155
```

6.1 K-Means Applied to Pitch Estimates

K-means was selected as a candidate classification model because of its relative simplicity. K-means uses euclidean distance between each of the vectors to classify the data into k groups of the shortest distance. The groups are then used to remap the centroids to the mean of each group, and the groups are calculated again. This process continues until an update causes no change in the group assignments, at which point the model has converged.

6.1.1 Performance of K-Means

The model was checked using Leave One Out Cross Validation (LOOCV). All but one `wav` files were used in the model. The result was predicted on the left-out `wav` file. This process was repeated until every file had been left out and the subsequently tested against. Because multiple events may exist per file, several were included in each validation of the model.

6.2 Support Vector Machine Applied to Pitch Estimates

A Support Vector Machine (SVM) is a machine learning supervised model that attempts to separate classes of data using a training set and a given method. SVM is able to perform non-linear classification by using

[2] This is true within reason; we don't want parents to become complacent because of too many false positives

hyperplanes and so may be more suited to the classification task than K-means.

SVM relies on a set kernel and cost for solving a model. The kernel dictates what hyperplanes SVM can and will use in it's modeling. In soft SVM, the cost variable dictates the "cost" or punishment of the model for violating the classifications during the training phase. Alternatively, hard SVM will fail to missclassify, but may also fail to converge for that reason. This data was not linearly separable, so SVM was unable to sort the sounds exactly, but a reasonable hyperplane was drawn.

The SVM model was run in `R` using the `e1071` package[3].

6.2.1 Performance of SVM

The model was checked using LOOCV in the same style as was done for K-means (see 6.1.1 for details). Multiple tests were performed with various kernels (sigmoid, polynomial and linear), but the radial basis kernel was found to perform the best. Using the radial basis kernel, we varied the misclassification cost and ran the model, producing a plot of Power and Accuracy at various costs.

6.3 Using Baum-Welch Algorithm

7 Conclusion

8 Shortcomings

9 Recommendations

[3] `http://cran.r-project.org/web/packages/e1071/e1071.pdf`