UNIVERSITY OF CALIFORNIA, LOS ANGELES

Cry, Cry, Cry

Kyle Colton
Christian Gao
Benjamin Hong
Feiran Zhu

Contents

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 Description of Data

For this project, we were given 183 `wav` files split between several sounds:

19	Adult Female Cry
15	Adult Female Laugh
15	Adult Male Laugh
15	Adult Female Neutral
15	Adult Male Neutral
15	Animal Cat
15	Animal Dog
21	Baby Cry
18	Baby Laugh
25	Baby Neutral

Each file is approximately 1.5 seconds in length and contains lossless sound without background noise. Each file is labeled according to the contents.

4 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 4.1, 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 $t = 1.40$.

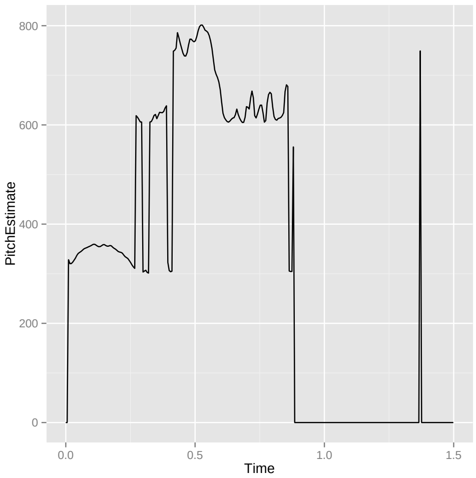

Figure 4.1: Pitch estimate of a `wav` file

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 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 4.2: 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.

```
> 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
```

Figure 4.3: Summary statistics produced from the Aubio output

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

4.1 Measuring Results

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.

Total Population	Baby is Crying	Baby is not Crying
Predicted Crying	True Positive (TP)	False Positive (FP)
Predicted Not Crying	False Negative (FN)	True Negative (TN)

The false positive (FP) is also referred to as Type 1 Error. Similarly, the false negative is referred to as Type 2 Error. The main statistics explored for ranking models were:

- Overall prediction rate (or accuracy):

$$ACC = \frac{\sum TP + \sum TN}{\sum \text{Total Population}}$$

- Power (also known as Sensitivity or True Positive Rate (TPR)):

$$TPR = \frac{\sum TP}{\sum TP + \sum FN}$$

- Precision (also known as the Positive Predictive Value (PPV)):

$$PPV = \frac{\sum TP}{\sum TP + \sum FP}$$

In simple terms, the accuracy is the fraction of total stuff that was correctly classified. Power is the number of times we caught the baby crying over the number of times it did cry. Precision is the fraction of times we thought the baby was crying where the baby was actually crying.

4.2 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.

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

4.2.1 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.

While the overall prediction rate of k-means grew over time, the power decayed. The best option for a value of k is therefore $k = 2$. However, the performance at $k = 1$ suggests that there is not a very clear-cut grouping to choose. Therefore, we should expect that a more complex model may perform better. K-means is a fast algorithm and is easy to understand, but the reliance on euclidean distances hampers the performance.

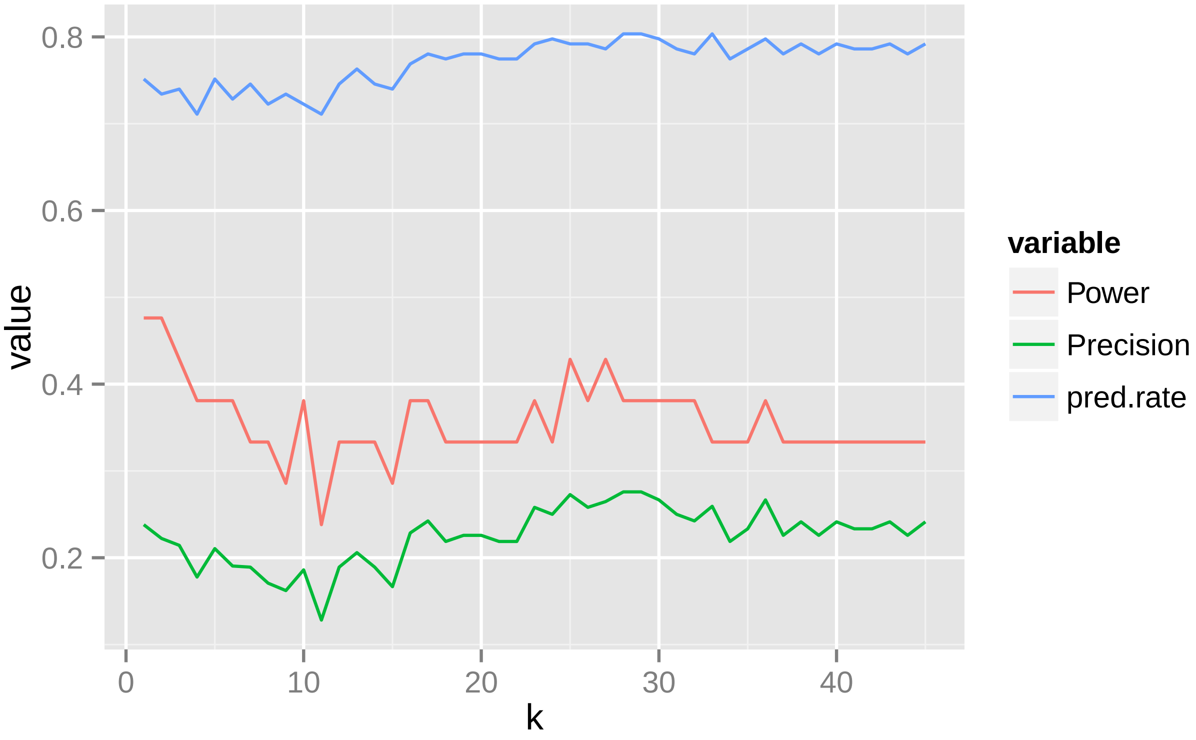

Figure 4.4: Performance K-means for various values of *k*

4.3 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 multidimensional planes and shapes 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].

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

4.3.1 Performance of SVM

The model was checked using LOOCV in the same style as was done for K-means (see 4.2.1 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.

Power initially starts very low, but rises as cost increases to approximately $cost = 25$. Once we maximize power, the best prediction rate appears at around $cost = 27$.

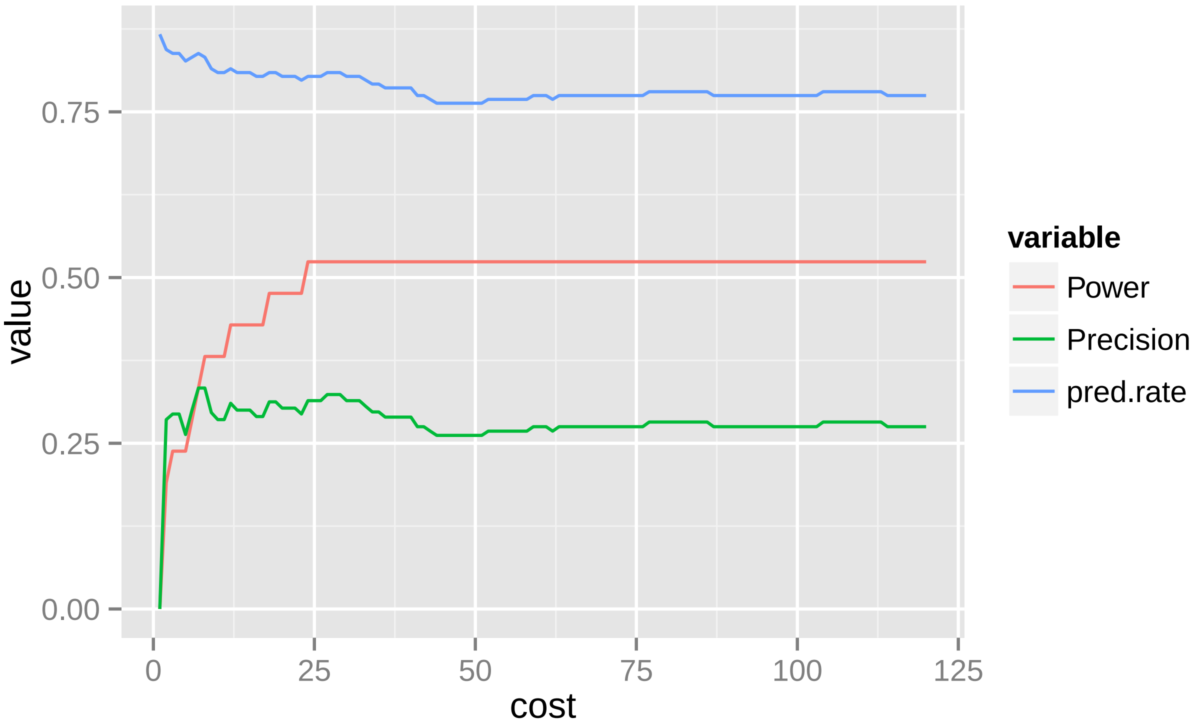

Figure 4.5: Performance of SVM versus misclassification cost

4.4 Comparison of K-Means and SVM

In this experiment, we saw slight benefits in using SVM over K-Means. Using 2 clusters, K-Means achieved a prediction rate of approximately $ACC = 0.734$, a power of $TPR = 0.476$, and a precision of $PPV = 0.222$. At $cost = 27$, SVM saw a prediction rate of $ACC = 0.809$, a power of $TPR = 0.524$, and a precision of $PPV = 0.324$. SVM was able to correctly classify more often, and achieve a higher power, catching slightly over half of all the baby cries.

While K-Means has likely reached its limits with this Data, we would expect SVM to continue to improve given more data. Unfortunately, SVM is generally very computationally expensive compared to K-Means. With the increased complexity, though comes better modeling capabilities.

4.5 Future Improvements

Improvements can be made by including other measurement points, for example, relative volume. We would expect, for example that a baby crying and a baby laughing would have distinctly different volumes. Combining both volume and pitch analysis would likely improve the model.

Additionally, we could expect that given a larger dataset, the classification would improve. This is especially true if SVM is allowed to train on a particular baby's sounds.

5 Hidden Markov Model

Old Transition Matrix

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

↓

New Transition Matrix

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

The third approach we tried is to apply Hidden Markov Model to our crying baby data in order to get simulated prediction results. As we decided to use pitch as the predictor and classified it into four categories–high pitch, medium pitch, low pitch, and no pitch, we first need to extract pitches from the WAV file. We designed an algorithm to get the maximum pitch at every point in time, smooth the max points into an envelope pitch curve, and separate the max pitch curve at the troughs. The resulted separation plot is shown. The second step was to get parameters for the Hidden Markov Model. We got the emission matrix directly by summarizing the original data. For the transition matrix, however, we did not have a reliable basis and had to arbitrarily make the transition probability of crying to laughing or to making neutral sounds. Then we applied Viterbi algorithm to find the most likely combinations in the sequence of pitches we extracted.

5.1 Using Baum-Welch Algorithm

Our simulated results shows an around 40% percent success rate of determining whether the baby is crying or laughing. Honestly, this success rate is not compelling due to the many constraints in our data and model which we would further discuss shortly.

6 Conclusion

Ultimately, we ended up with three models, two of which performed fairly well (K-means and SVM), and one that, with a more comprehensive dataset may yield better results.

7 Shortcomings

We encountered many problems when applying the Hidden Markov Model in the prediction. The small sample size led us to one of our first problems. Since we only have 62 WAV files related to baby sounds, we could not get a reliable estimation for the initial emission matrix used in the hidden Markov model. Given the fact that the accuracy of hidden Markov model heavily relies on the emission matrix and transition matrix, the lack of a reliable emission matrix compromises our success rate. That the length of each audio recorded is really short was also problematic for us. Most of the recorded sounds are less than 2 seconds. It is relatively hard to determine whether a baby is crying or laughing with just one sound lasting for only one second. If we could get a substantially large sample with each recording lasting for more than one hour, we are confident that our simulated results would be much more reliable. Another shortcoming of our model is that we could not get a convincing transition matrix. The transition matrix lists the probability that a cry sound would stay crying, or change to either laugh or neutral in the next state. We arbitrarily determined the transition probabilities in our model, and this matrix might be significantly different from the real transition probabilities.

8 Recommendations

Though the algorithm based on the hidden Markov model only performed slightly better than a random number generator, we believe that with a more comprehensive data set, we would be able to yield more appealing results. Ideally, the data set would consist of hours and hours of long recordings of baby noise and have over 10,000 observations of individual baby noises. With such a data set in hand, we would be able to answer the following questions (and therefore conduct a more thorough analysis using HMM):

- What is the probability that a baby will cry right after laughing?
- When does the baby usually sleep?
- What is the probability that a baby cries right after waking up?
- What is the average time that a baby cries, laughs and is neutral?