2017

# Abnormal Heartbeat Sound Detection

MUHAMMAD ASIM

**Abnormal Heartbeat Sound Detection**

# 1. Introduction

The goal of this project is to train a classifier on heartbeat sounds so that it can detect abnormal heartbeats and classify patient's cardiac physiology accordingly. The scope of this project is implement a classification method by training on data collected from various clinical institutes from different populations: CinC challenge dataset 2016.

Most, cardiac patients are not diagnosed early. This is due to many reasons: patient's negligence, patients only start to show symptoms when it is late. In a country like Pakistan where the number of doctors are just not enough and some are not even skilled e.g. in rural areas, there is an urgent need for such a project that can detect abnormalities based on heartbeat sounds.

We want to classify between two types of audio: normal and abnormal heartbeat sounds. Heartbeats are composed of alternating 'lub' and 'dub' sounds, also called S1 and S2. In normal category, there are normal, healthy heart sounds. Abnormal heartbeat sounds can be further divided into 'murmur' and 'extra systole'. Murmur sounds have "whooshing" turbulent noises between alternating S1 and S2. 'Extra systole' heart sounds can be identified from an additional heartbeat. Both murmur and extra systole heartbeats can be symptoms of heart disorders. Early detection of these conditions could help a person get proper treatment in time and in some cases save lives.

The problem is twofold: feature extraction from audio files and secondly to classify heart sound from the features extracted. We are using CinC Challenge 2016 data set which contains random length audio files recorded clinically. These audio files contain noise thus making this task really challenging, because it has to work for noisy samples also. We will be extracting perceptual features[1] and MFCC[1] from audio files and use them to train an ensemble of support vector machines classifier[2].

# 2. Problem Statement

1. For any audio signal $\psi$, design a de-noising method $D$, that can remove noise and also exaggerate heartbeat sounds, where $\upsilon$ is the de-noised audio signal of length $N$.
$$\upsilon = D(\psi)$$

2. For any audio signal $\upsilon$ of length $N$, design a feature extraction method $f$ that can extract a feature vector $\boldsymbol{x} \in \mathbb{R}^d$ from it.
$$\boldsymbol{x} = f(\upsilon)$$
3. Design a classification method $h$ to classify audio files into normal $y = 0$ or abnormal $y = 1$ from their extracted features.
$$y = h(\boldsymbol{x})$$
4. Since the dataset is imbalanced, the classifier $h$ must be designed in such a manner that is robust against the class imbalance.

## 3. Dataset: CinC Challenge 2016

The dataset contains more than 3000 audio files collected from different research groups from different countries over a period of more than a decade. The dataset contains audio files from different population groups. The dataset also contains noisy audio recording thus adding authenticity to the dataset[4]. Detailed description of the dataset can be found in [3].

The dataset has a class imbalance between normal and abnormal recordings. Normal class audio recordings are nearly 4 times abnormal class audio recordings. This class imbalance presents a challenging task[5] to correctly predict abnormal heartbeat.

## 4. Proposed Method

This section describes the final proposed methodology of the project.

**De-noising**

Since the dataset contains noisy audio recordings, the first task is to remove the noise and pre-process the audio files so that features can be extracted from them. In [6], the author de-noises heartbeat audio files using wavelet transform, low pass filtering and smoothing, and also finds the locations of S1 and S2.

After inspecting audio recordings, it was determined that in the process of recording, extra noise is introduced while placing and remove the digital stethoscope. To eliminate this noise, audio signals were truncated by 0.4s from either side.

According to [6], most information is expected to lie in low frequencies whereas only noise resides on higher frequencies. Following [6], the truncated audio signals were decomposed using wavelet transform into approximate and detailed parts. The details were completely removed and the audio signal was reconstructed only from the approximate parts. Fourth-level order six daubechies filter was employed for decomposing the audio signals. The reconstructed audio signals were low pass filtered at 195Hz because we expect that heartbeat frequency spectrum to lie inside this frequency at all times. The filtered signals were then triangle smoothed using [7].

A peak detection algorithm was employed to find S1 and S2 positions. A new approach was introduced here. Empirical evaluations showed that this algorithm was just as good as [6]. It should be noted that S1-S2 locations were ultimately not used as a features in classification. The algorithm can be found in the file 'S1_S2_SegmentLoc.m'.

## Feature Extraction

The second task is to extract features from these processed audio files. In [1], the author uses perceptual features along with Mel-Frequency Cepstral Coefficients to form a feature vector $\boldsymbol{x}$ for classification purpose. To this end, we divided an audio signal $v$ into frames of 64ms. These frames were then hamming windowed. We identified the non-silent frames using the same process as in [1], but with a threshold value of 0.085, which was determined empirically. Following [1], Perceptual features and MFCCs were extracted from these non-silent frames as shown below:

1. **Total Spectrum Power**

$$P = \log\left(\int_0^{f_0} |\mathcal{F}(f)|^2 \, df\right)$$

Where $f_0/2$ is the sampling frequency.

2. **Sub-band Powers**
The frequency spectrum was divided into 4 sub-bands with intervals $[0, f_0/8]$, $[f_0/8, f_0/4]$, $[f_0/4, f_0/2]$, and $[f_0/2, f_0]$. Log sub-band powers were computed for these frequency bands.

$$P_j = \log\left(\int_{L_j}^{H_j} |\mathcal{F}(f)|^2 \, df\right)$$

3. **Brightness**

$$f_c = \left( \frac{\int_0^{f_0} f \, |\mathcal{F}(f)|^2 df}{\int_0^{f_0} |\mathcal{F}(f)|^2 \, df} \right)$$

4. **Bandwidth**

$$B = \sqrt{\frac{\int_0^{f_0} (f - f_c)^2 \; |\mathcal{F}(f)|^2 df}{\int_0^{f_0} |\mathcal{F}(f)|^2 \, df}}$$

5. **Mel-Frequency Cepstral Coefficients**
   19 MFCC coefficients were computed for non-silent frames.

The mean and standard deviations of these features were computed to form a feature vector $\boldsymbol{x} \in \mathbb{R}^d$ , where $d = 53$. It should be noted that implemented my own entire feature extraction code in Matlab from scratch.

## Classification

This section explains our final classification strategy after rigorously testing different algorithms. According to [2], Ensemble Support Vector Machines are one of the many strategies to deal with imbalanced data. It works by dividing the majority class training dataset into smaller $m$ training subsets such that each majority class training subset is of the same as the minority class training dataset. Now, $m$ SVM classifiers are trained with each majority class training subset against minority class training dataset. Final prediction is performed using majority voting scheme. Fig.1 is an illustration of how training of Ensemble SVM works.

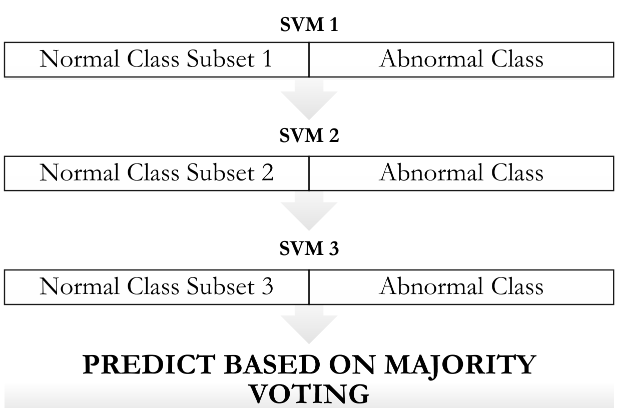


Figure 1: Ensemble SVM

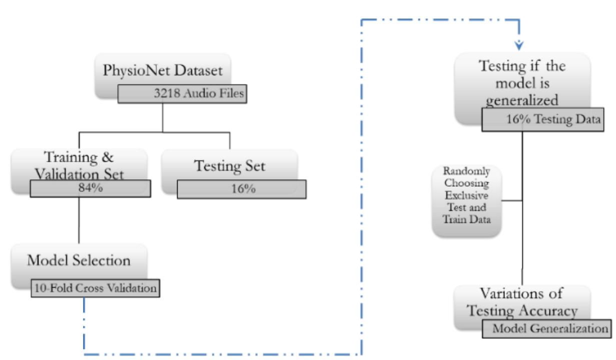


Figure 2: Training, Validation and Testing Methodology

It should be noted that training, validation and testing of Ensemble SVM was implemented from scratch because Matlab does not have in-built support for such algorithms. The training, validation and testing scheme is illustrated in the fig. 2.

Grid search was used to find the optimal values of hyper-parameters for our model through 10-fold cross validation. The hyper parameters include '$m$', '$boxconstraint$', '$kernel\ function$' and its '$kernelscale$'. It should be noted that since the normal class is four times the abnormal class it makes sense to test for only three values of $m = 3,4,5$. The kernel function and other hyper parameters' values are stated in the Table 1 for all three SVM classifiers.

| Ensemble SVM | Kernel Function | C | Kernel Scale | m |
|---|---|---|---|---|
| SVM 1 | RBF | 100 | 7.015 | 3 |
| SVM 2 | RBF | 100 | 7.015 | |
| SVM 3 | RBF | 100 | 7.015 | |

Table 1: Model Selected through 10-fold cross validation

Using grid search, it was empirically concluded that same hyper-parameters for all three SVMs resulted in best unbiased validation accuracy. The final results are mentioned in section 5.

## 5. Experimental Results

As shown in fig 2, the training data was divided into fixed training data and testing data. The training data was used for training, validation and model selection. The validation accuracy of Ensemble SVM using the optimal values of hyper parameters determined by using 10-fold validation accuracy is **82.66%**. This model was then tested on the testing data and confusion matrix in fig 3 shows that not only test accuracy was very close to validation accuracy, but also balanced in its classification accuracy.

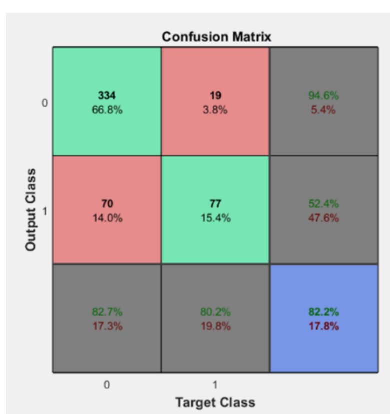


Figure 3: Confusion Matrix for Test Data

The confusion matrix in fig. 3 suggest that our model does perform well for unknown data. But it does it generalize well. One could argue that, the test data was easy. To test the generality of our model, we split our original dataset into randomly selected training and testing set: we randomly choose exclusive training and testing data as illustrated in fig 2. The results of this experiment are shown in fig 4.


Figure 4: Testing the Model for Generality

Fig. 4 confirms that the testing accuracy over different test data remains almost constant. Hence, we are now in position to claim that our model would also generalize well to the new data. Our proposed technique is comparative to the state of the art as mentioned in Section 7.

## 6. Other Experiments Conducted

Many experiments and models were tested before we choose Ensemble SVM as our final model. Most notably these include: SMOTE, ADASYNC, Cluster SVM and Neural Networks.

**Cluster SVM**

As explained in [2], in Cluster SVM the majority class is divided into $m$ clusters such that each cluster is of the same size as the minority class. Then $m$ SVMs are employed just like in Ensemble SVM for training and prediction. The majority class is divided into $m$ clusters using k-means clustering. This scheme had some drawbacks:

1. Although more balanced accuracy was achieved between true positive and true negative accuracy, but overall accuracy was greatly reduced.
2. Clusters generated were not of same size as the minority class. Experiments were done by clustering data into 3 and 5 clusters. Table 4 presents the different cluster sizes against minority class size.
3. Classification was in fact biased towards minority class.

<table>
<tr><th>Minority Class Size</th><th>m</th><th>Clusters</th><th>Cluster Size</th></tr>
<tr><td rowspan="8">657</td><td rowspan="3">3</td><td>1</td><td>1519</td></tr>
<tr><td>2</td><td>638</td></tr>
<tr><td>3</td><td>404</td></tr>
<tr><td rowspan="5">5</td><td>1</td><td>240</td></tr>
<tr><td>2</td><td>1176</td></tr>
<tr><td>3</td><td>400</td></tr>
<tr><td>4</td><td>301</td></tr>
<tr><td>5</td><td>444</td></tr>
</table>

Table 2

**SMOTE**

Synthetic Minority Over Sampling Technique as mentioned in [5], is a very powerful algorithm that has produced great results. It was worthwhile to apply this technique to our problem. First, 500 test data samples were separated from the dataset. Normalizing parameters: mean and variance where calculated from the remaining training samples. The abnormal minority class was then up sampled using SMOTE[9]. The new training data size is mentioned in the table 2.

| | |
|---|---|
| Minortiy Class Training Data Size | ~ 1800 |
| Majority Class Training Data Size | ~ 2200 |

Table 3

10-Fold cross validation was employed on SVM for model selection. The validation accuracy as mention in table 3 suggested very promising results.

| 10-fold Cross Validation Accuracy | SVM Kernel | Hyper Parameters | |
|---|---|---|---|
| 97% | RBF | C = 100 | Kernel Scale = 7.01 |

Table 4

But on testing, it was concluded that the model was biased towards abnormal heart sounds. It showed poor accuracy for normal audio heartbeat sounds. The testing scheme illustrated in fig 2 was employed and test results are shown in figure 5.


Figure 5: Testing Results of SMOTE

As is evident from confusion matrices in fig 5, we could not consider SMOTE as our final approach. The root of the problem in this technique was that we generate synthetic data, which did not give a full reflection on how it would perform during testing.

**ADASYN**

Adaptive Synthetic Sampling[5] is an extension of SMOTE. SMOTE populates the minority synthetically from within. ADASYN enforces the synthetic samples to populate the boundary between two classes. The algorithm[10] was tested to investigate the prospects of this technique. Since, intial resutls were not encouraging, we did not pursue this technique any further.

### Neural Network

We implemented a two layer neural network with sigmoid and tanh activation functions. Initial investigation showed a validation accuracy of above 90%, but with abnormal class accuracy of lesser than 80%. Since the goal of the project was to achieve good accuracy for abnormal detection all the while not compensating on the overall accuracy too much, neural networks were not investigated further.

## 7. Current State of the Art

This project was a follow up on CinC Challenge 2016. According to [4], the best accuracy of 86% was achieved using Adaboots & CNN. A detailed table from [4], shows the top 8 scores and their methodology.

| Rank | Entrant | *Se* | *Sp* | *MAcc* | Method note |
|---|---|---|---|---|---|
| 1 | Potes *et al.* | 0.9424 | 0.7781 | 0.8602 | AdaBoost & CNN |
| 2 | Zabihi *et al.* | 0.8691 | 0.8490 | 0.8590 | Ensemble of SVMs |
| 3 | Kay & Agarwal | 0.8743 | 0.8297 | 0.8520 | Regularized Neural Network |
| 4 | Bobillo | 0.8639 | 0.8269 | 0.8454 | MFCCs, Wavelets, Tensors & KNN |
| 5 | Homsi *et al.* | 0.8848 | 0.8048 | 0.8448 | Random Forest + LogitBoost |
| 6† | Maknickas | 0.8063 | 0.8766 | 0.8415 | Unofficial entry - no publication |
| 7 | Plesinger *et al.* | 0.7696 | 0.9125 | 0.8411 | Probability-distribution based |
| 8 | Rubin *et al.* | 0.7278 | 0.9521 | 0.8399 | Convolutional NN with MFCs |
| 17† | Voting of top N=38 algorithms | 0.7120 | 0.9015 | 0.8068 | Simple mode |
| 43† | Sample entry | 0.6545 | 0.7569 | 0.7057 | See section 3 |

Table 5

## 8. Future Work

This project only investigated neural network with a glance. Future work hopes to investigate deep learning techniques for sound classification of abnormal heartbeat sound detection. Relevant works by Andrew Y. Ng[11] and in [12] on feature extraction from sound signals will be investigated, because machine learning algorithms can be only as good as the features used.

## References


1. Guo, Guodong, and Stan Z. Li. "Content-based audio classification and retrieval by support vector machines." IEEE transactions on Neural Networks 14.1 (2003): 209-215.
2. Batuwita, Rukshan, and Vasile Palade. "Class imbalance learning methods for support vector machines." Imbalanced learning: Foundations, algorithms, and applications 83 (2013).
3. Liu, Chengyu, et al. "An open access database for the evaluation of heart sound algorithms." Physiological Measurement 37.12 (2016): 2181.
4. Clifford, Gari D., et al. "Classification of normal/abnormal heart sound recordings: The physionet/computing in cardiology challenge 2016." Proceedings of the Computing in Cardiology (2016): 609-612.
5. He, Haibo, and Edwardo A. Garcia. "Learning from imbalanced data." IEEE Transactions on knowledge and data engineering 21.9 (2009): 1263-1284.
6. Deng, Yiqi, and Peter J. Bentley. "A robust heart sound segmentation and classification algorithm using wavelet decomposition and spectrogram." Extended Abstract in the First PASCAL Heart Challenge Workshop, held after AISTATS. 2012.
7. https://www.mathworks.com/matlabcentral/fileexchange/19998-fast-smoothing-function
8. Chawla, Nitesh V., et al. "SMOTE: synthetic minority over-sampling technique." Journal of artificial intelligence research 16 (2002): 321-357.
9. https://www.mathworks.com/matlabcentral/fileexchange/38830-smote--synthetic-minority-over-sampling-technique
10. https://www.mathworks.com/matlabcentral/fileexchange/50541-adasyn--improves-class-balance--extension-of-smote-
11. Lee, Honglak, et al. "Unsupervised feature learning for audio classification using convolutional deep belief networks." Advances in neural information processing systems. 2009.
12. Aytar, Yusuf, Carl Vondrick, and Antonio Torralba. "Soundnet: Learning sound representations from unlabeled video." Advances in Neural Information Processing Systems. 2016.