Classification of Cardiac Arrhythmias Patients

Azar Fazel, Fatema Algharbi, Batool Haider

CS229 Final Project Report, Fall 2014

Abstract:

Cardiac Arrhythmias are any of a group of conditions in which the electrical activity of the heart is irregular or is faster or slower than normal. It is the leading cause of death for both men and women in the world [1][2]. In this project, we aim to classify heart arrhythmias patients among 16 different classes based on ECG (Electrocardiography) data. After applying rigorous data pre-processing and feature selection techniques, we used 5 different machine learning algorithms; SVM, Logistic Regression, KNN, Random Forest and Decision Trees. Our best accuracy was 73%, obtained via SVM. We also used some of these methods to come up with the most important attributes that determined the class of arrhythmia. This work can be of immense importance to researchers who are exploring various techniques to capture the key pre-informers of a potential cardiac disease, well before it is too late.

Introduction

Heart diseases kill more than 385,000 people annually. In the United States, someone has a heart attack every 34 seconds [4]. In this paper, we present our methodology and the outcomes of developing a machine learning system that is capable of classifying a patient into 16 different cardiac arrhythmic categories. This work has great potential to serve the medicine industry. With the advancement in medical technology, database will only continue to grow. The evolution of smart body chips capable of sending real time patients' information are rigorously being researched upon. Algorithms such as these can be have ground breaking impact regarding helping researchers target the keys features that cause cardiac arrhythmia and assist them in classifying patients in right categories, so as to be able to take measures in the right direction.

Data

The data has been taken from a well-maintained ECG (Electrocardiography) database (https://archive.ics.uci.edu/ml/datasets/Arrhythmia). It contains 279 attributes (ECG/ Patient related variables) and 452 instances. The variable 'Class' is our target variable. Class 01 refers to 'normal' ECG, classes 02 to 15 refer to different classes of Arrhythmia and class 16 refers to the rest of unclassified classes. Figure (1) shows the distribution of different classes in our database. As can be seen, almost half of the instances are classified to class 1, while there are few instances in other classes. Thus, in the database, we do not have much evidence for some of the classes like class 02 or 03.

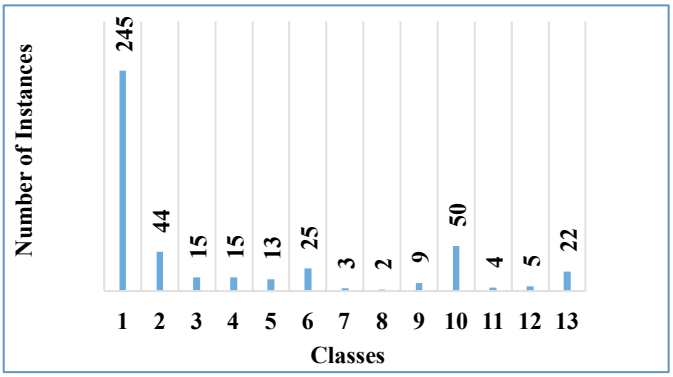

Fig. 1. Distribution of instances in various classes

Analysis & Data Pre-processing

Following are the main steps took to process the data.

1) Normalization

Since many algorithms require normalized dataset, we normalized all the feature values except "gender" using Z-score ($z = \frac{x-\mu}{\sigma}$) normalization and scaled them between 0 and 1. We then proceeded on towards analyzing the data.

2) Feature Selection:

We first ran various algorithms on the data set with all the features. The performance of most was not

satisfactory. With the instances to feature ratio of only 1.6, we decided to improve the performance of algorithms by reducing the number of features.

i) Invariant Features

Firstly, we removed some the categorical features that were 95% of time indicating either all 0's or all 1's. These seemed not to help a lot in decision making as they were pointing to the same category most of the time. As expected, their removal, did not disturb the accuracy but reduced the feature space and simplified the data base.

ii) Mutual information:

Next, we found tuples of features that were correlated at least with 0.95 correlations. This is to say the set of features giving approximately the same information, and so we included only one feature from each tuple. This reduced the number of features to 160.

iii) Feature Importance Score via Decision trees:

We took two approaches: the first was to cluster the classes into "Class 1" and "Not Class 1". We then found the features that best described the responses for these two large classes. In the second approach, we used all the classes in their original labeling in the dataset and found out the most important features for the response via decision tree. Figure 2.1 shows the results of decision tree. As seen in this figure, based on the features importance score, only 11 features out of 160 features were considered as the most important features.

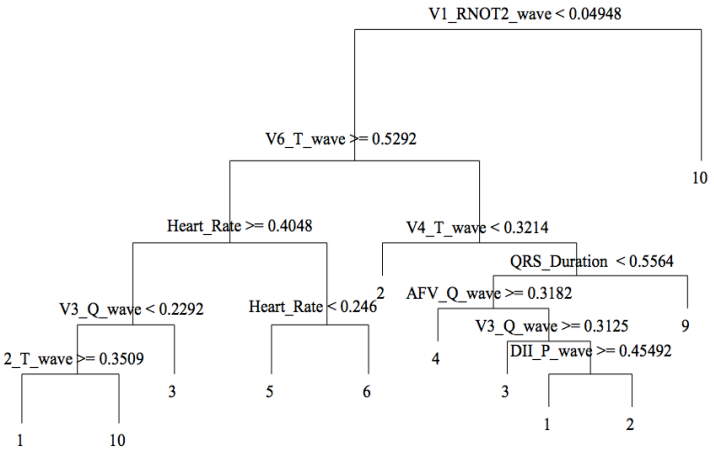

Fig. 2.1 Decision Tree for Cardiac Arrhythmias patients

Figure 2.2 shows the most important 30 variables for the response using the decision trees. We noticed that the heart rate is the most important variable followed by V6_T_wave, V5_T_wave and V3_Q_wave.

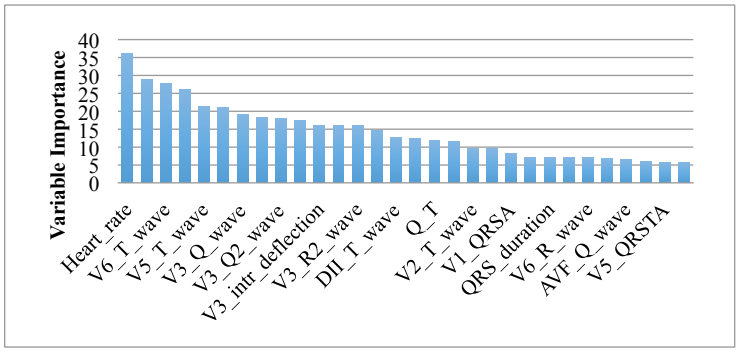

Fig. 2.2 Variable Importance for Cardiac Arrhythmias patients

We then applied Principle Components Analysis (PCA) to these features to reduce the dimensionality of the dataset while retaining most of the variation in the data set. Figure 3. shows the results of this analysis.

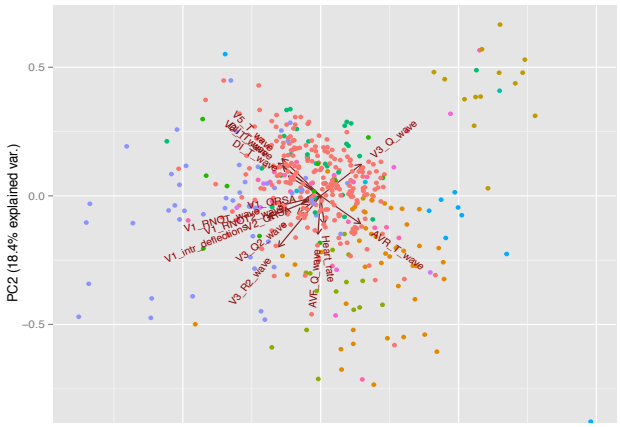

Fig. 3. Using PCA to visualize the features selected by Decision Tree

Models:

We applied several models and algorithms to our dataset, accounting for their merits and demerits based on literature review. These are as follows:

1) KNN (K-Nearest Neighbors)

We used KNN because it is simple to implement & very straight forward. Here, an object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors [3]. This could be done by measuring 'distances' between the object and its neighbors. The following formula shows a representation of simple Euclidian distance, where 'a' and 'b' are the respective positions of the object and one of its neighbors

$$D(a,b) = \sqrt{\sum_{i}^{n}(b_i - a_i)^2}$$

KNN is very sensitive to irrelevant or redundant features because all features contribute to the similarity and thus to the classification. This was ameliorated by careful feature selection described previously.

2) Decision Trees

We used Decision trees as they implicitly perform feature selection & can tackle nonlinear relationships between parameters. Each leaf of the tree is labeled with a class or probability distribution over the classes.

A tree can be "learned" by splitting the source set into subsets based on attributes and the recursion is completed when splitting no longer adds value to the predictions. The information gained is based on the decrease in "entropy" after dataset is split. Following equation shows the formula for entropy, where 'p' is the probability of certain class occurring, given a specific feature

$$E(s) = -p(P)\log 2p(P) - p(N)\log 2p(N)$$

3) Random Forest

We tried Random forests as they are an ensemble learning method that operate by constructing multitude of decision trees. Therefore their performance is often better than decision trees alone and can tackle issues like 'pruning' (often an issue in decision tree) automatically. Since random forest works quite well with several features, we first tried it on the full set of features. Table 1 shows its performance class wise.

Class	N Cases	N Instances misclassified	Pct. Error
1	245	192	78.37%
2	44	24	54.55%
3	15	1	6.67%
4	15	1	6.67%
5	13	6	46.15%
6	25	12	48.00%
7	3	3	100.00%
8	2	2	100.00%
9	9	1	11.11%
10	50	29	58.00%
14	4	1	25.00%
15	5	2	40.00%
16	22	21	95.45%

Table 1. Random Forest Performance on full feature set

Since the error obtained was quite high (approx. 50%), we then tried it on the reduced features set outlined previously. The error graph (figure 4) below shows that the performance did not improve much.

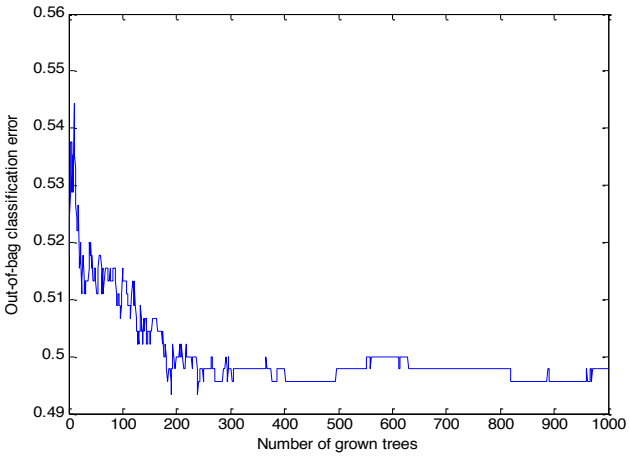

Figure 4. Random Forest Performance on reduced feature set

4) SVM (Support Vector Machines)

We used SVM with a cross validation of k = 10 to allow us average the error, among the 10 accuracy we found, the highest that was reached is 0.8 and the mean was approximately 0.73.

We tried both the polynomial and the linear kernels for the SVM and found out that the linear kernel outperformed the polynomial kernel. The linear SVM with CV = 10 gave the best accuracy among all the other models we used. The features that were selected for the SVM where determined by using the decision trees as explained in the feature selection section.

4) Logistic Regression

Since the logistic regression is used for binary classification of datasets with categorical dependent features, in order to apply logistic regression to our multi-class dataset, we firstly classified our instances into two major classes, class 1 (which contained all the instances with "class 01" label) and class NOT-1 (which contained the instances for all the other classes). We classified our data in this way, because about half of our instances were labeled as class 01.

Just like SVM, we used cross validation with k=10 folds to validate our model. Although we got accuracy of about 0.92 for the training set, the accuracy for test set was about 0.62.

Results

Table 2 summarizes the results obtained from each of the outlined methods.

As can be seen SVM with linear kernel gave the best performance. Trees on the other hand did not perform so well. One of the possible reasons may be the presence of 16 classes. Our discussion with some of the other CS229 teams with similar project revealed that decision trees performance improved drastically with reduction in the total number of classes. Other methods appear to be less sensitive to this.

Additionally, we used Cross Validation to obtain error estimate on the test set. This helped us to be sure that the error was the mean of all the various test sets that could be obtained from the given data and so results are less sensitive to the choice of test/training set. We used 10 k-folds for the purpose. Figure 5 shows the comparison between errors that we got for different models.

Model	Training Accuracy	Test Accuracy	Parameters
Support Vector Machine (SVM)	0.812	0.727	CV: k = 10
K-Nearest Neighbor (KNN)	0.684	0.600	K neighbors = 10 CV: k = 10
Logistic Regression for Class 1 and Not Class 1	0.925	0.629	CV: k = 10
Random Forests	-	0.53	2/3 Training Set
Decision Trees	0.784	0.356	CV: k = 10

Table 2. Summary of various models' performance using cross validation of 10 folds

Figure 5 shows the accuracy for the SVM, KNN, Decision Trees, and Logistic Regression using various folds. It is clear that the SVM outperformed all of the other models in our case.

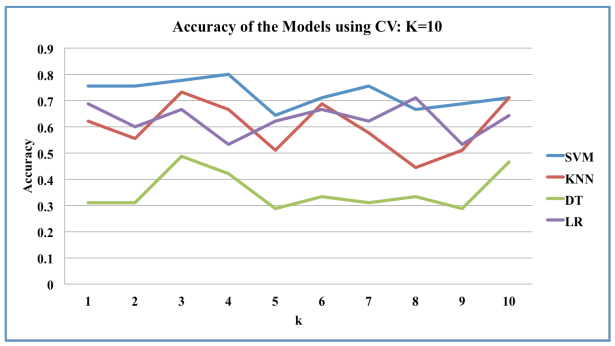

Figure 5. Comparison of the accuracy of various models using cross validation

Future Work

Since about 50% of the data were clustered in class 1, it is believed that with more data on the patients of the other classes, it is possible to learn more to get more accurate classification. Some of the classes had only 2 to 3 instances in the data; which makes it difficult to learn about these classes and hence their misclassification's probability is high when using various algorithms. It is clear that class 1 has the dominant effect on the predicting models so collecting more instances of patients in the other classes is a goal for better predictions in future.

Apart from this and based on our findings throughout this project, here is what we would like to propose for future work in this area-

1. It will be interesting to group the features based on their physical similarity (like also the ECG's P wave variables together and all the Q wave variables together) and re-check the performance of these algorithms.
2. Though we used some rigorous feature selection techniques, one more method that can be tried is "Forward" or "Backward" search techniques, where the features are dropped or added to check their impact on the algorithms' accuracy. A great thing to do here would be to combine all the feature selection techniques described in this paper and average out the score assigned to each feature in the data set. This will give researchers a very good idea of which features are most important distinguisher of various Cardiac Arrhythmias. These ranks could be discussed with experts in the field of Cardiology to check how well did the data driven assignments match expert opinion.

3. It will be worthwhile to reduce the number of classes based on literature review (group similar classes together) to check if the performance of the model improves. One could break this algorithm into two parts. First part can be used to give the user results based on reduced number of classes (say 5) and then on all 16 classes. This way, even if the accuracy with all the 16 classes is not extremely high, at least the user will know in

general which of the 5 broad categories he/she fall in.

Conclusions

Using linear SVM, we built a predicting model to classify the Cardiac Arrhythmia Patients. The most important features were selected via decision trees. We were successful in reducing the variables from 279 to 15 variables and using cross validation our models accuracy is approximately 73%. The data is skewed as about 50% of the patients are classified as class 1; hence, for prediction accuracy improvement it is essential to gather more data on patients in other classes.

References

[1].http://en.wikipedia.org/wiki/Cardiac_dysrhythmia
[2].http://www.cdc.gov/dhdsp/data_statistics/fact_sheets/docs/fs_heart_disease.pdf
[3].Cunningham 2007. k-Nearest Neighbor Classifiers. Technical Report UCD-CSI-2007-4. University College Dublin
[4] Roger VL et al. Heart disease and stroke statistics—2012 update: a report from the American Heart Association http://www.cdc.gov/Other/disclaimer.html