Prediction and Classification of Cardiac Arrhythmia

Pammi Sairam, Akhilesh Yadav, Samujjal Das, Rajesh Kumar Jha

Dept. of CSE, Indian Institute of Technology Guwahati

CS 561 Project Report

***Abstract* -- Cardiac Arrhythmia[1] is a group of conditions in which the electrical activity of the heart is irregular or is faster or slower than normal. Many heart arrhythmias are harmless: however, if they are particularly abnormal, or result from a weak or damaged heart, arrhythmias can cause serious and even potentially fatal symptoms[2]. It is the leading cause of death for both men and women in the world[3].**

In this project, we plan to predict Cardiac Arrhythmia[11] based on a patient's medical record. Our primary objective is to classify a patient into one of the Arrhythmia classes like Tachycardia and Bradycardia based various measurements like his ECG measurements and help us in understanding the application of machine learning in medical domain. After appropriate feature selection we plan to solve this problem by using Machine Learning Algorithms namely Support vector machine, Softmax Regression, Naïve Bayes, Decision Tree, Adaptive Boosting (AdaBoost) and Random Decision Forest.

I. Introduction

The total number of deaths due to cardiovascular diseases[4] read 17.3 million a year according to the WHO causes of death. Thus, how to predict cardiac arrhythmia in real life is of great significance[5]. We have chosen this project since arrhythmia has no symptoms and is a well-known silent killer, so it can be a boon for people who do not have any clue at their disposal regarding their medical condition. In this project, we plan to develop a machine learning system that can classify a patient into different cardiac arrhythmic classes.

The diagnosis of cardiac arrhythmia can be classified into various classes based on the Electrocardiogram(ECG) readings and other attributes. First class will refer to the normal patient while other classes shall represent different classes of cardiac arrhythmia like Tachycardia, Bradycardia and Coronary artery diseases. This is a supervised learning problem.

2. Data Set

The dataset[8] for the project is taken from the UCI Repository https://archive.ics.uci.edu/ml/datasets/Arrhythmia[6] There are (452) rows, each representing medical record of a different patient. There are 279 attributes like age, weight and patient's ECG related data. Out of 279 attributes, 206 are linear valued and the rest are nominal. General attributes like age and weight have discrete integral values while other ECG features have real values. The dataset provides a base for training and testing different models used and furthermore setting a benchmark for the same.

The aim is to distinguish between the presence and absence of cardiac arrhythmia and to classify it in one of the 13 groups. Class 01 refers to 'normal' ECG classes 02 to 12 refers to different classes of arrhythmia and class 13 refers to the rest of unclassified ones. The 13 classes are as following -

Number	Class	Instances
1	Normal	245
2	Ischemic changes (Coronary Artery)	44
3	Old Anterior Myocardial Infarction	15
4	Old Inferior Myocardial Infarction	15
5	Sinus tachycardia	13
6	Sinus bradycardia	25
7	Ventricular Premature Contraction (PVC)	3
8	Supraventricular Premature Contraction	2
9	Left bundle branch block	9
10	Right bundle branch block	50
11	Left Ventricle hypertrophy	4
12	Atrial Fibrillation or Flutter	5
13	Others	22

Table 1: Different Classes of Arrhythmia.

3. Survey

The aim is to distinguish between the presence and absence of cardiac arrhythmia and to classify it in one of the 13 groups. For the time being, there exists a computer program that makes such a classification. However, there are differences between the cardiology's and the programs classification. Taking the cardiology's as a gold standard we aim to minimize this difference by means of machine learning tools.

4. SCOPE

These machine learning techniques can be deployed in hospitals where a large dataset is available and can help the doctors in making more precise decisions and to cut down the number of causalities due to heart diseases in the future.

Furthermore, the same can be used and employed in wearable devices that can track the patients' conditions and report depending on the complexity. Moreover, on the unavailability of doctors the machine can handle the situation for a given amount of time.

5. METHODOLOGY

A. Feature Selection[20]*:*

Firstly, we removed some of the categorical features that were 95% of time indicating either all 0's or all 1's. If any training instance has a missing value for a given attribute, we set it as the mean of the value plus or minus the standard deviation for that attribute related to the class it belongs to. The central premise when using a feature selection technique is that the data contains many features that are either redundant or irrelevant, and can thus be removed without incurring much loss of information.

If for a given attribute majority of values are missing, then we discard that attribute and remove it from our training set. The features can be grouped into five blocks –

- features concerning biographical characteristics, i.e., age, sex, height, weight and heart rate.
- features concerning average wave durations of each interval (PR interval, QRS complex, and ST intervals).
- features concerning vector angles of each wave.
- features concerning widths of each wave.
- features concerning amplitudes of each wave.

B. Principal Component Analysis[16]*:*

PCA[7] is being used to identify patterns in the data and then expressing the data in such a way to highlight similarities and differences. Primarily we are using PCA to reduce the number of dimensions by identifying the more important features i.e. the principal components. The number of principal components is less than or equal to the smaller of the number of original variables. The first principal component has the largest possible variance and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components.

The main idea of principal component analysis (PCA) is to reduce the dimensionality of a data set consisting of many variables correlated with each other, either heavily or lightly, while retaining the variation present in the dataset, up to the maximum extent. The same is done by transforming the variables to a new set of variables, which are known as the principal components (or simply, the PCs) and are orthogonal, ordered such that the retention of variation present in the original variables decreases as we move down in the order. So, in this way, the 1st principal component retains maximum variation that was present in the original components. The principal components are the eigenvectors of a covariance matrix, and hence they are orthogonal.

C. Random Forests and Decision Trees[9]*:*

Decision trees are a type of model used for both classification and regression. Trees answer sequential questions which send us down a certain route of the tree given the answer. both columns.

We implement a Random Forest classifier. The model works by continually sampling with replacement a portion of the training dataset, and fitting a decision tree to it. The number of trees refer to the number of times the dataset is randomly sampled. Moreover, in each sampling iteration, a random set of features are selected. In decision trees, each node refers to one of the input variables, which has edges to children for all possible values that the input can take. Each leaf corresponds to a value of the class label given the values of the input variables represented by the path from the root node to the leaf node. The number of trees and the number of leaves are learned via cross validation.

Random forests are a strong modelling technique and much more robust than a single decision tree. They aggregate many decision trees to limit overfitting as well as error due to bias and therefore yield useful results.

5. MODELS

A. *Naïve – Bayes Classifier*[18]*:*

It is a classification technique based on Bayes Theorem[19] with an assumption of independence among predictors. Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes' theorem with the "naive" assumption of independence between every pair of features.

Using the formula of conditional probability, we have the probability of whole training set:

$$P(x,y) = \prod_{i=1}^{m}\left(\prod_{j=1}^{n}\phi_{j,x_j^{(i)}|y=y^{(i)}}\right)\phi_{y^{(i)}}$$

We implemented our own Naive Bayes binomial and multinomial classifiers in Python. We use Naive Bayesian equation to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of prediction.

Naive Bayes learners and classifiers can be extremely fast compared to more sophisticated methods. The decoupling of the class conditional feature distributions means that each distribution can be independently estimated as a one-dimensional distribution. This in turn helps to alleviate problems stemming from the curse of dimensionality.

B. *SVM*[17] *(Support Vector Machines):*

Support Vector Machines are based on the concept of decision planes that define decision boundaries. A decision plane is one that separates between a set of objects having different class memberships.

Here we aim to minimize the following equation:

$$\left[\frac{1}{n}\sum_{i=1}^{n} max\left(0,1-y_i\left(\underset{w}{\rightarrow}.\underset{x_i}{\rightarrow}-b\right)\right)\right]+\lambda\ \|\underset{w}{\rightarrow}\|^2$$

where the parameter λ determines the trade-off between increasing the margin-size and ensuring that the $\underset{x_i}{\rightarrow}$ lie on the correct side of the margin.

In SVM, a hyperplane is selected to best separate the points in the input variable space by their class, either class 0 or class 1. In two- dimensions you can visualize this as a line. You can make classifications using this line. By plugging in input values into the line equation, we calculate whether a new point is above or below the line. We tried both the polynomial and the linear kernels for the SVM and found out that the linear kernel outperformed the polynomial kernel, and have implemented linear kernel leaving the polynomial one.

C. *Softmax Regression*[10]*:*

Here we use a generalised version of logistic regression that is Softmax regression. This can be used for multi-class classification (Provided classes are mutually exclusive). In logistic regression number of classes was binary but here it can take any number of classes. We just here replace the sigmoid function of logistic regression with so called Softmax function:

$$P\left(y=j|z^{(i)}\right)=\phi_{softmax}\left(z^{(i)}\right)=\frac{e^{z^{(i)}}}{\sum_{i=0}^{k}e_k^{z^{(i)}}}$$

In the Softmax regression setting, we are interested in multi-class classification (as opposed to only binary classification), and so the label y can take on K different values, rather than only two. Thus, in our training set {(x(1), y(1)),…,(x(m) , y(m))}, we now have that y(i) ∈{ 1,2,…,K }. (Here the convention that we follow is, the classes will be starting from 1, rather than from 0.)

For an example, in our case we have 13 categories to classify arrhythmia in, so here the value that K will take is 13. For different cases the value of K changes. For the purpose of digit recognition, we will have the value of K to be 10 for ten digits to be recognised.

Here we have provided with the training cost of Softmax regression. The training accuracy was same for both the splits that is for 80%-20% and 70%-30%. The learning rate chosen was .00001.

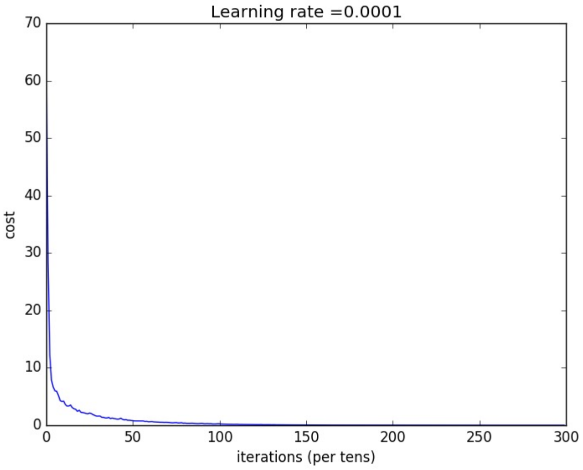

Image 1: Softmax Regression training error rate.

D. *Decision Tree*[13]*:*

We use a non-parametric supervised method used for classification called Decision Trees, it predicts the value of a target variable by learning simple decision rules inferred from the data features.

A decision tree typically starts with a single node, which branches into possible outcomes. Each of those outcomes leads to additional nodes, which branch off into other possibilities. This gives it a treelike shape.

There are three different types of nodes: chance nodes, decision nodes, and end nodes. A chance node, represented by a circle, shows the probabilities of certain results. A decision node, represented by a square, shows a decision to be made, and an end node shows the final outcome of a decision path. Decision tree proved to be a powerful method and provided us with an accuracy more than that of the average accuracy.

E. *AdaBoost*[14] *(Adaptive Boosting):*

AdaBoost[15] refers to a particular method of training a boosted classifier. A boosted classifier can be of the form:

$$F_T(x)=\sum_{t=1}^{T} f_t\,(x)$$

where each f_t is a weak learner that takes an object x as input and returns a value indicating the class of the object.

Ada-boost classifier combines weak classifier algorithm to form strong classifier. A single algorithm may classify the objects poorly. But if we combine multiple classifiers with selection of training set at every iteration and assigning right amount of weight in final voting, we can have good accuracy score for overall classifier.

After training a classifier at any level, AdaBoost assigns weight to each training item. Misclassified item is assigned higher weight so that it appears in the training subset of next classifier with higher probability.

After each classifier is trained, the weight is assigned to the classifier as well based on accuracy. More accurate classifier is assigned higher weight so that it will have more impact in final outcome.

F. *Random Forests*[12]*:*

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.

The training algorithm for random forests applies the general technique of bootstrap aggregating or bagging to tree learners. Given the training set with responses bagging repeatedly selects a random sample with replacement of the training set and fits trees to these samples.

6. Results

The main objective of this project was to develop a system that could robustly detect an arrhythmia. The second objective of this project was to develop a method to robustly classify an ECG trace into one of 13 broad arrhythmia classes. We report our performance for each of the six methods using one methodology. Principal component analysis has been used for all of them.

We show results for each algorithm, as well as vary other parameters for better results. The following are two tables with results. In the first case, the training-testing data was split 80% - 20% and in the second case, the training-testing data was split 70% - 30%.

Model	Training Accuracy	Test Accuracy
Naïve-Bayes Classifier	55.57%	54.52%
Support Vector Machine	100%	65.83%
Softmax Regression	100%	66.26%
Decision Tree	85.12%	71.32%
Adaptive Boosting	100%	67.65%
Random Forests	100%	70.92%

Table 2: For 80% - 20% split.

Model	Training Accuracy	Test Accuracy
Naïve-Bayes Classifier	54.63%	50.73%
Support Vector Machine	100%	70%
Softmax Regression	100%	60.62%
Decision Tree	87.39%	72.63%
Adaptive Boosting	100%	78.16%
Random Forests	100%	72.52%

Table 3: For 70% - 30% split.

7. Conclusion

It is clear from the above data that the SVM and Softmax Regression algorithms are capable of automatically detecting arrhythmias with reliable accuracy (Training Data = 100 % and Testing Data=65%). Furthermore, Random Forests consistently performs better than PCA in terms of feature selection. Our general approach in this project was as follows. We started with Random Forests and we tried to obtain maximum accuracy but we got around 71% by splitting the data into 80% training and 20% testing, which clearly indicated lack of training data Then we used Softmax Regression which uses the softmax function and we ran it using Adam optimizer. Softmax regression gave comparatively poor results with average accuracy around 66 %. Naïve-Bayes classifier gave poor results due to problem of lack of enough training examples (452) and excessive number of features. Adaptive boosting Using PCA gave the best results with average accuracy of classification around 100 % for training set and 78 % for testing set.

8. Acknowledgement

We are highly grateful to our professor Dr. Rashmi Dutta Baruah for her continued guidance and support throughout the course of this project.

9. 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] http://www.heart.org/HEARTORG/Conditions/Arrhythmia/Arrhythmia_UCM_002013_SubHomePage.jsp

[4] Roger VL et al. Heart disease and stroke statistics— 2012 update: a report from the American Heart

[5] http://www.texasheart.org/HIC/Topics/Cond/bbblock.cfm

[6] http://archive.ics.uci.edu/ml/

[7] https://www.dezyre.com/data-science-in-python-tutorial/principal-component-analysis-tutorial

[8] https://archive.ics.uci.edu/ml/datasets/Arrhythmia

[9] https://towardsdatascience.com/decision-trees-and-random-forests-df0c3123f991

[10] http://ufldl.stanford.edu/tutorial/supervised/SoftmaxRegression/

[11] https://www.encyclopedia.com/medicine/diseases-and-conditions/pathology/arrhythmia

[12] https://en.wikipedia.org/wiki/Random_forest

[13] https://en.wikipedia.org/wiki/Decision_tree

[14] https://en.wikipedia.org/wiki/AdaBoost

[15] https://machinelearningmastery.com/boosting-and-adaboost-for-machine-learning/

[16] https://en.wikipedia.org/wiki/Principal_component_analysis

[17] https://en.wikipedia.org/wiki/Support_vector_machine

[18] https://en.wikipedia.org/wiki/Naive_Bayes_classifier

[19] http://www.saedsayad.com/naive_bayesian.htm

[20] https://en.wikipedia.org/wiki/Feature_selection