Prediction and Classification of Cardiac Arrhythmia

Vasu Gupta, Sharan Srinivasan, Sneha S Kudli
{*gvasu, sharanms, skudli*}*@stanford.edu*

Abstract **- Cardiac Arrhythmia refers to a medical condition in which heart beats irregularly. This paper aims to detect and classify arrhythmia into 14 different variants. A few popular techniques from contemporary literature were implemented namely Naive Bayes, feature selection, SVM, Random Forests and Neural Networks.A new approach combining SVM and Random Forests classifiers was also implemented.**

1 Introduction

Irregularity in heart beat may be harmless or life threatening. Hence both accurate detection of presence as well as classification of arrhythmia are important. Arrhythmia can be diagnosed by measuring the heart activity using an instrument called ECG or electrocardiograph and then analysing the recorded data. Different parameter values can be extracted from the ECG waveforms and can be used along with other information about the patient like age, medical history, etc to detect arrythmia. However, sometimes it may be difficult for a doctor to look at these long duration ECG recordings and find minute irregularities. Therefore, using machine learning for automating arrhythmia diagnosis can be very helpful. The project aims at using different machine learning algorithms like Naive Bayes, SVM, Random Forests and Neural Networks for predicting and classifying arrhythmia into different categories.

2 Data Set

The dataset for the project is taken from the UCI Machine Learning Repository `https://archive.ics.uci.edu/ml/datasets/Arrhythmia` (1 csv file, 1 information file).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.

The data set is labeled with 16 different classes. Classes 2 to 15 correspond to different types of arrhythmia.Class 1 corresponds to normal ECG with no arrhythmia and class 16 refers to unlabeled patient.The data set is heavily biased towards the no arrhythmia case with 245 instances belonging to class 1 and 185 instances being split among the 14 arrhythmia classes and the rest 22 are unclassified. 3 of the classes related to the degree of AV block do not appear in the data set. The labels for this data set are obtained from cardiologists and they are considered to be the gold model.

The main challenges in processing this data set are the limited number of training examples compared to the number of features, heavy bias towards the case of normal ECG, missing feature values (about 0.33%) and feature values belonging to both continuous and categorical types.

3 Data Preprocessing

The original data contains columns with both missing values and single valued columns having the same value for all the patient records.These columns were deleted from the data set. The resulting data set contained 452 instances and 257 features.

4 Feature Selection

We experimented with two different filter feature selection techniques. One of the reasons for using fewer features was the limited number of data records(452) compared to 257 features. This helps in avoiding overfitting and also gives insight into the important features which have maximum correlation with the output lables but minimal correlation among themselves.

In the first technique, we discretized all the continuous valued columns and then computed the mutual information I(Y,X) between each feature and the output label vector using the below formula 1(H refers to en-

tropy). The scores (mutual information value) obtained for each feature were then normalised to remove any biases that appeared due to discretization of the real valued columns. This normalisation technique is suggested in [3]. Features with higher scores were considered more important. In this approach, we did not compute the correlations between the features themselves. This technique was implemented in Matlab.

$$I(Y,X)(=score) = H(Y) - H(Y|X)$$
$$score := \frac{2 * score}{H(Y) + H(X)} \quad (1)$$

Our second approach was to use a matlab feature selection package named mRMR `http://featureselection.asu.edu/software.php`. This technique selects the features which have both maximum correlation with the output labels and minimum correlation among themselves. It also uses some advanced techniques(Weka package) `http://www.cs.waikato.ac.nz/ml/weka/` for discretizing the real valued columns. Therefore, we used the results from this second approach while implementing SVM and Random Forests. The corresponding error versus number of selected features' curves are shown in figures 2 and 4 respectively. The top features extracted have column numbers near 105 to 110 and 235 to 240 which correspond to average width and amplitude respectively of Q,R,S,etc waves in channel V2 of ECG recordings.

5 Models and Results

The following subsections discuss and provide results obtained with Naive Bayes, feature selection, SVM, Random Forests, Neural Networks and fusion of these different techniques.

5.1 Naive Bayes Classifier

We implemented our own Naive Bayes binomial and multinomial classfiers in Matlab. This implementation was performed without any feature reduction. Results obtained are given in tables 1 and 2. Many of the feature are real valued and so these were discretised individually into different levels. The results shown are with 30 different discretisation levels. We aslo experiemented with different number of discretisation levels from 20 to 60 but the test errors were almost similar. Results shown are for two different cases. In the first one, the training-testing data was split 70% - 30% and 3 fold cross validation was performed. In the second case, the training-testing data was split 80% - 320% and 5 fold cross validation was performed. All the features were used to train the model. Both the test and train errors are high, indicating that Naive Bayes is not able to capture the data distribution effectively. Ineffective feature discretisation may also be a contributing factor.

Table 1: Naive Bayes Binomial Classification

Train-Test set size	Test error	Train error
70%-30%	31.12	27.32
80%-20%	31.86	27.03

Table 2: Naive Bayes Multinomial Classification

Train-Test set size	Test error	Train error
70%-30%	52.94	39.55
80%-20%	52.96	43.21

5.2 SVM

SVM is effective in high dimensional spaces like the arrhythmia data set. First, mRMR feature selection was performed. The data set was then split into 70%-30% between train and test respectively. Since we are dealing with a skewed data set with small number of rows, we employed bootstrapping to improve the performance of the algorithm.The train data was doubled in size using random sampling, while making sure all the data points in the original train data were represented atleast once.

To determine the type of kernel most apprropriate, the SVM model was built using polynomial kernels of varying degrees and a guassian kernel. The quadratic kernel, resulted in a good model fit, minimizing the generalization error as can be seen in Figure 1. This led us to the inference that there were significant second order interactions among the feature variables in the design matrix.

Figure 2 plots the generalization accuracy on the test set with the number of top features selected. It can be seen that the best accuracy is obtained with around 254 features.

The Confusion Matrix in Figure 3 shows that Class 1 and 5 were often being confused for one another. The SVM model was unable to predict Class 16, which is primarily believed to be because of the inherent ambiguity of the class (i.e Class 16 refers to a state of uncer-

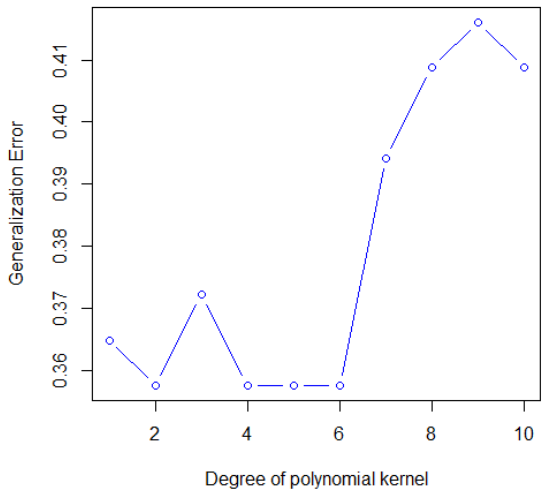

Figure 1: Generalization Error with Polynomial Kernal Degree for Multinomial SVM

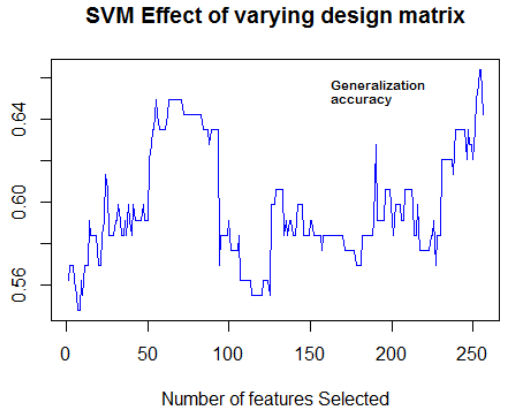

Figure 2: SVM Generalization Accuracy with Number of Features Selected

	Predicted												
Truth	1	2	3	4	5	6	7	8	9	10	14	15	16
1	53	5	0	2	3	3	1	0	0	2	0	0	1
2	4	9	1	0	0	0	0	0	0	0	0	0	0
3	0	0	5	0	0	0	0	0	0	0	0	0	0
4	0	0	0	1	0	0	0	0	0	0	0	0	0
5	3	0	0	1	2	0	0	0	0	0	0	0	0
6	3	0	0	0	0	2	0	0	0	0	0	0	0
7	0	0	1	0	0	0	0	0	0	0	0	0	0
8	0	1	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	3	0	1	0	0
10	3	1	1	0	0	2	0	0	0	12	0	0	0
14	1	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	1	0
16	6	1	0	0	0	0	0	0	0	2	0	0	0

Figure 3: Confusion Matrix SVM with polynomial degree 2 kernel

tain cardiac activity). Additionally, due to the highly bised distribution of classes, the model proved inefficient in predicting classes with low density. In specific both classes 7 and 8 saw only one tuple in the test set. The sheer lack of data, meant that there was no way to build meaningful distributions of the features needed to classify classes 7 and 8. To address the issue of misclassifying class 5 (sinus Tachycardia) as class 1 (normal), we used an anomaly detector.

Anomaly detection - We treated the SVM as a one class classifier and separated all the data points from the origin (in feature space F) in order to maximize the distance from this hyperplane to the origin. This results in a binary function which captures regions in the input space where the probability density of the data lives. Thus the function returns +1 in the region (capturing the training data points) and 1 elsewhere. On finding anomalies in the data set, we used our intuitive reasoning from the SVM confusion matix viz. that class 5s were mostly misclassified as 1s. Hence we found the anomalies which lied far from the data set, closest to the origin and detected the points predicted by our SVM model as 5. We reclassified these states of possible sinus tachycardia as normal state. This helped improve the accuracy to 70%

5.3 Random Forests

A simple decision tree gives good predictions when there is a huge number of predictor variables like in the this data set. Early methods to construct decision trees were unstable with small perturbations in data resulting in large changes in predictions. Random forests is an ensemble classifier that consists of many decision trees and outputs the class that is the mode of the classes output by individual trees. In this way, an RF ensemble classifier performs better than a single tree.

Figure 4 shows that the generalization error botttoms out beyond selecting around top 50 features.

We followed the implementation in [2] and obtained similar results. The data was split as shown in Figure 5 as train-test 80-20 % respectively. Figure 6 shows the training error for a single run. Figure 7 shows the training error for a single run with the simple random sampling (SRS) approach detailed in the [2]. SRS reduces the bias in the class distribution and has the same effect as bootstrapping.

Finally, the Random Forest technique was applied with 70-30 % data split and bootstrapping as explained in the SVM section 5.2. The resulting confusion matrix

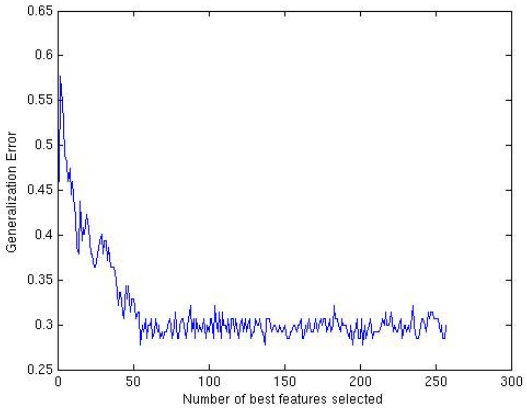

Figure 4: Random Forests Accuracy with number of top features

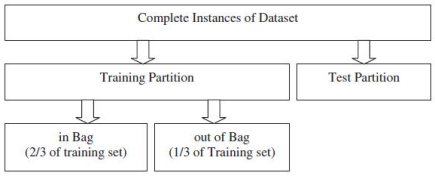

Figure 5: Illustration of Data Set Division for RF

is shown in Figure 8. Overall generalization accuracy was 72.3%

5.4 Fusion of SVM and Random Forests

SVM with polynomial kernal of degree 2 and RF method corresponding to 8 gave similar conf matrix w.r.t classes 5,6,10,11. However a linear kernel SVM which gives a lower overall accuracy classifies classes 5,6,10,11 better. We believe the reason for this could be because the separating hyperplane for classes 5,6,10,11 was linear and the quadratic kernel was not able to segregate data space this way. Hence we used a serial classfier consisting of RF and linear kernal SVM which gave us a generalization error of 22.6% or accuracy of 77.4%. The confusion matrix is as shown in Figure 9.

5.5 Hierarchical RF Classifier

We also tried a new approach with random forest classification where instead of one we train two different RF classifiers, the first one provides a binary classification about whether the person has arrhythmia or not. Then we further sub-classify the instances which are predicted with arrhythmia using the second random forest classifier. Using this approach, we obtained 20% generalisa-

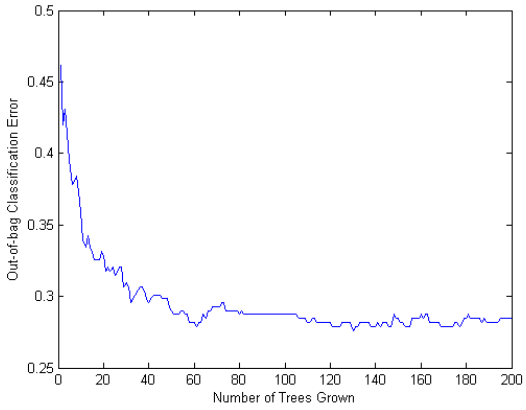

Figure 6: Classfication error for a single run of treeBagger without SRS

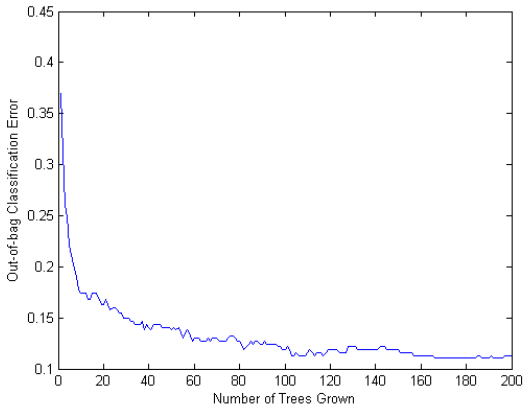

Figure 7: Classfication error for a single run of treeBagger with SRS

	Predicted												
Truth	1	2	3	4	5	6	7	8	9	10	14	15	16
1	64	5	0	0	0	0	0	0	0	1	0	0	0
2	1	13	0	0	0	0	0	0	0	0	0	0	0
3	0	0	5	0	0	0	0	0	0	0	0	0	0
4	0	0	0	1	0	0	0	0	0	0	0	0	0
5	3	1	0	0	1	0	0	0	0	1	0	0	0
6	4	0	0	0	0	0	0	0	0	1	0	0	0
7	0	0	1	0	0	0	0	0	0	0	0	0	0
8	1	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	3	0	0	0	1
10	6	1	0	0	0	0	0	0	0	12	0	0	0
14	1	0	0	0	0	0	0	0	0	0	0	0	0
15	0	1	0	0	0	0	0	0	0	0	0	0	0
16	6	1	0	0	0	0	0	0	0	2	0	0	0

Figure 8: Random Forests Confusion Matrix

	Pred												
truth	1	2	3	4	5	6	7	8	9	10	14	15	16
1	64	5	0	0	0	0	0	0	0	1	0	0	0
2	1	13	0	0	0	0	0	0	0	0	0	0	0
3	0	0	5	0	0	0	0	0	0	0	0	0	0
4	0	0	0	1	0	0	0	0	0	0	0	0	0
5	2	1	0	0	2	0	0	0	0	1	0	0	0
6	2	0	0	0	0	2	0	0	0	1	0	0	0
7	0	0	1	0	0	0	0	0	0	0	0	0	0
8	1	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	3	0	0	0	1
10	3	1	0	0	0	0	0	0	0	15	0	0	0
14	1	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	1	0
16	6	1	0	0	0	0	0	0	0	2	0	0	0

Figure 9: RF + SVM Confusion Matrix

tion error for just binary classification and 30% error for combined mutli classification. The error in this case is slightly more than what we obtain with a single multi class random forest classifier. This is probably because the overall accuracy is limited by the accuracy of the first level binary classifier. This technique could be improved further by using the SRS strategy.

5.6 Neural Networks

We used pattern net from the neural netwrok toolbox in Matlab to distinguish between the 16 classes. Pattern recognition networks are feedforward networks that can be trained to classify inputs according to target classes. This gave a classfication accuracy of 69%.

6 Conclusion

The paper presents the implementation of a few techniques used by contemporary papers on the arrhythmia data set. We also implemented a serial classifier using a fusion of linear kernel SVM and RF which gave us a generalization error of 77.4%. This provides a marginal improvement over the generalization errors reported by the papers we surveryed. The results are summarised in Table 3

7 Future Work

A number of combinations of algorithms can be implemented in the hierarchical scheme. Currently, we have implemented one 2-level scheme with RF. We can expand this to add more levels and try it with other models. Also for the Network implementation, we think bootstrapping may help improve the performance.

Table 3: Results Summary for 16 class classification

Algorithm	Test Accuracy(%)
Naive bayes	47
SVM-poly deg 2	66
RF	72
RF + SVM	77.4
Pattern Net	69
2 level RF	70

References

[1] H. Altay Guvenir, Burak Acar, Gulsen Demiroz, Ayhan Cekin "A Supervised Machine Learning Algorithm for Arrhythmia Analysis." *Proceedings of the Computers in Cardiology Conference, Lund, Sweden*, 1997

[2] zift, Akin."Random forests ensemble classifier trained with data resampling strategy to improve cardiac arrhythmia diagnosis." *Computers in Biology and Medicine* **41.5** (2011): 265-271

[3] Hall, Mark A., and Lloyd A. Smith. "Feature Selection for Machine Learning: Comparing a Correlation-Based Filter Approach to the Wrapper." *FLAIRS conference*. 1999.

[4] Uyar, Asl, and Fikret Gurgen. "Arrhythmia classification using serial fusion of support vector machines and logistic regression." Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, 2007. IDAACS 2007. 4th IEEE Workshop on. IEEE, 2007.

[5] Polat, Kemal, Seral ahan, and Salih Gne. "A new method to medical diagnosis: Artificial immune recognition system (AIRS) with fuzzy weighted pre-processing and application to ECG arrhythmia." Expert Systems with Applications 31.2 (2006): 264-269.

[6] Rudokait-Margeleviien, Dovil, Henrikas Pranevi-ius, and Mindaugas Margeleviius. "Data classification using Dirichlet mixtures." Information Technology and Control 35.2 (2006): 157-166.