



BOOTSTRAP APPROACH FOR ENVISAGE HEART ATTACK DISEASES USING COMBINE AD-BA ALGORITHM

D.Richard*

Dept of Computer Science
Research Scholar, Bharathiar University, India
revanrichard@gmail.com;

Dr.R.Mala

Dept of Computer Science
Alagappa University, India

Manuscript History

Number: **IRJCS/RS/Vol.05/Issue04/APCS10095**

Received: 03, April 2018

Final Correction: 13, April 2018

Final Accepted: 19, April 2018

Published: April 2018

Citation: D.Richard & Dr.R.Mala (2018). BOOTSTRAP APPROACH FOR ENVISAGE HEART ATTACK DISEASES USING COMBINE AD-BA ALGORITHM. IRJCS:: International Research Journal of Computer Science, Volume V, 193-202. [doi://10.26562/IRJCS.2018.APCS10095](https://doi.org/10.26562/IRJCS.2018.APCS10095)

Editor: Dr.A.Arul L.S, Chief Editor, IRJCS, AM Publications, India

Copyright: ©2018 This is an open access article distributed under the terms of the Creative Commons Attribution License, Which Permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited

Abstract— In worldwide, heart disease is the leading reason of passing away many peoples. One in every four people is afflicted with and dies of heart disease. Early and exact diagnoses of heart disease thus are critical in improving the chances of long term survival for patients and saving millions of lives. In this research paper, an advanced collection of genetic approach are utilizing an Adaptive and Boosting algorithm(AD-BA), is developed for accurate coronary heart disease diagnosis and result predictions

Keywords— Genetic approach; Bootstrap model; AD-BA algorithm; voting interface;

I. INTRODUCTION

Globally, heart disease is the leading cause of death for both men and women [1], with more than half of the deaths occurring in men. One in every four people is afflicted with and dies of heart disease, [1].Heart disease encompasses several types of heart conditions. The most common type of heart condition is coronary heart disease [1]. In this research, an alternative and enhanced genetic approach is proposed for heart disease prediction based on classification and prediction merge an adaptive Boosting algorithm that merge a set of classifiers into a well-built collection learning prediction model. Genetic methods are rough set approach and fuzzy set and bootstrap model [2]. In general, these models are less commonly used for classification in medical data mining systems than the methods their strength in certain applications, and hence it is useful to include them here Genetic methods are easily parallelizable and have been used for classification as well as other optimization problems[2].

II. METHODOLOGY

Genetic approach attempt to incorporate ideas of natural evolution. In general, genetic learning starts as follows. An initial population is created consisting of randomly generated rules. Each rule can be represented by a string of bits. As a simple example, suppose that samples in a given training set are described by two Boolean attributes, A1 and A2, and that there are two classes, C1 and C2. The rule "IF A1 ANDNOT A2 THENC2" can be encoded as the bit string "100," where the two leftmost bits represent attributes A1 and A2, respectively, and the rightmost bit represents the class. Similarly, the rule "IF NOT A1 AND NOT A2 THENC1" can be encoded as "001." If an attribute has k values, where $k > 2$, then k bits may be used to encode the attribute's values. Classes can be encoded in a similar fashion. Based on the notion of survival of the fittest, a new population is formed to consist of the fittest rules in the current population, as well as offspring of these rules. Typically, the fitness of a rule is assessed by its classification accuracy on a set of training samples. Offspring are created by applying genetic operators such as crossover and mutation. In crossover, substrings from pairs of rules are swapped to form new pairs of rules. In mutation, randomly selected bits in a rule's string are inverted. The process of generating new populations based on prior populations of rules continues until a population, P, evolves where each rule in P satisfies a pre specified fitness threshold. Genetic approach easily parallelizable and have been used for classification as well as other optimization problems. In data mining, they may be used to evaluate the fitness of other algorithms.

RESOURCES AND METHODS

A. Heart Disease Dataset

The research used bootstrap data mining technique on UCI heart data sets to provide the diagnosis results. The results from the approach were so promising. If further attempts are engaged in the application of Information Technology in diagnosing various diseases such as heart then efficient, timely and decent healthcare services will be realized.

B. Bootstrap model

The accuracy estimation methods mentioned above, the bootstrap method samples the given training tuples consistently with replacement. That is, each time a tuple is selected, it is equally likely to be selected again and re-added to the training set. There are several bootstrap methods. A commonly used one is the 63.2 % bootstrap, which works as follows. Suppose we are given a data set of d tuples. The data set is sampled d times, with replacement, resulting in a bootstrap sample or training set of d samples. It is very likely that some of the original data tuples will occur more than once in this sample. The data tuples that did not make it into the training set end up forming the test set. Suppose we were to try this out several times. As it turns out, on average, 63.2% of the original data tuples will end up in the bootstrap, and the remaining 36.8% will form the test set (hence, the name, .632 bootstrap). Each tuple has a probability of $1/d$ of being selected, so the probability of not being chosen is $(1-1/d)$. We have to select d times, so the probability that a tuple will not be chosen during this whole time is $(1-1/d)^d$. If d is large, the probability approaches $e^{-1} = 0.368$. Thus, 36.8% of tuples will not be selected for training and thereby end up in the test set, and the remaining 63.2% will form the training set. We can repeat the sampling procedure k times, where in each iteration, we use the current test set to obtain an accuracy estimate of the model obtained from the current bootstrap sample. The overall accuracy of the model is then estimated as

$$\text{Acc}(M) = \sum_{ai=1}^k (0.632 \text{ Acc}(M_i)_{\text{test set}} + 0.368 \text{ Acc}(M_i)_{\text{train set}})$$

C. Adaptive Boosting Algorithm

Ada boost assigns each training tuple an equal weight of $1/d$. Generating k classifiers for the ensemble requires k rounds through the rest of the algorithm. In round i, the tuples from D are sampled to proper training set, D_i , of size d. Sampling with replacement is used—the same tuple may be selected more than once. Each tuple's chance of being selected is based on its weight. A classifier model, M_i , is derived from the training tuples of D_i . Its error is then calculated using D_i as a test set. The weights of the training tuples are then adjusted according to how they were classified. If a tuple was incorrectly classified, its weight is increased. If a tuple was correctly classified, its weight is decreased. A tuple's weight reflects how hard it is to classify the higher the weight, the more often it has been misclassified. These weights will be used to generate the training samples for the classifier of the next round. The basic idea is that when we build a classifier, we want it to focus more on the misclassified tuples of the previous round. Some classifiers may be better at classifying some "hard" tuples than others.

In this way, we build a series of classifiers that complement each other. Now, let's look at some of the math that's involved in the algorithm. To compute the error rate of model M_i , we sum the weights of each of the tuples in D_i that M_i misclassified. That is,

$$\text{error}(M_i) = \sum_j w_j _err(X_j);$$

D. Bagging Algorithm

The bagging algorithm works as a method of increasing accuracy, for ease of explanation; assume at first that our model is a classifier. Suppose that the patient and would like to have a diagnosis made based on symptoms. Instead of asking on doctor, may choose to ask several. If a certain diagnosis occurs more than any of the others, you may choose this as the final or best diagnosis. Behind bagging. Intuitively, a majority vote made by a large group of doctors may be more reliable than a majority vote made by a small group.

E. Voting Method:

Increasing model accuracy: Bagging and Adaptive each generate a set of classification or prediction models, M_1, M_2, \dots, M_k . Voting strategies are used to combine the predictions for a given unknown tuple. Once boosting is complete, how is the ensemble of classifiers used to predict the class label of a tuple, X . The bagging, where each classifier was assigned an equal vote. That is, the final diagnosis is made based on a majority vote, where each doctor gets an equal vote. Now replace each doctor by a classifier, have the basic idea.

PRE-PROCESSES METHOD DIAGRAM

The pre processes method diagram the first steps to be process on input data set to set all the second step process on data set sub section in this process evaluate data to pass the bootstrap method to take on Adaptive boosting and bagging algorithm, this algorithm classify the data in proper way. The input features means to given data set inputs that is attributes, the attributes is age, sex, occupation, chest pain type (4 values) , food habits, resting blood pressure, fasting blood sugar > 120 mg/dl, resting electrocardiographic results (values 0,1,2), maximum heart rate achieved, old peak = ST depression induced by exercise relative to rest, the slope of the peak exercise ST segment, that: 3 = normal; 6 = fixed defect; 7 = reversible defect as follows 14 attributes as to be given input.



Fig1: Pre-Processes Method Diagram

METHOD COMBINES OF AD-BA ALGORITHM

In this combines of AD-BA Algorithm is get accuracy and with envisage, it's to genres previous predict algorithm to show the weighted result. To take intuitive look at how AD-BA Algorithm works as a method of increasing accuracy. For ease of explanation, we will assume at first that our model is a classifier. Suppose the patient and would like to have a diagnosis made based on symptoms. Instead of asking one doctor and may choose to ask several. If a certain diagnosis occurs more than any of the others to may choose this as the final or best diagnosis

III. PROPOSED ON COMBINES STRUCTURE OF AD- BA ALGORITHM

The increasing model accuracy Adaptive and Bagging algorithm each generate a set of classification and prediction models, M_1, M_2, \dots, M_k . Voting strategies are used to combine the predictions.

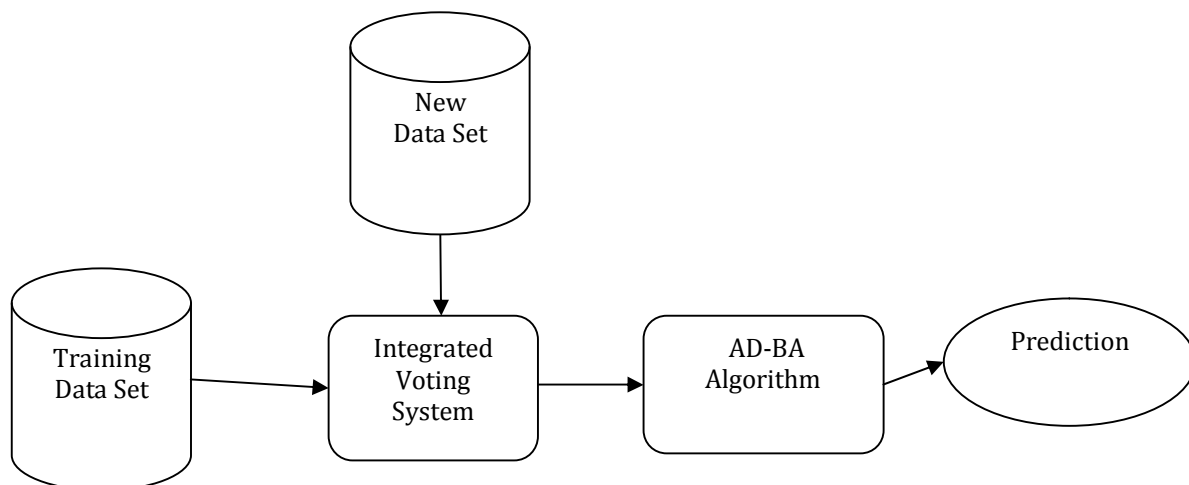


Fig2: Integrated Structure of AD-BA Algorithm

STEPS OF THE PROPOSED AD-BA ALGORITHM

Step 1: for $i = 1$ to k do // create k models:
Step 2: create AD-BA sample, D_i , by sampling D with replacement;
Step 3: use D_i to derive a model, M_i ;
Step 4: endfor
Step 5: To use the merged model on a tuple, X :
Step 6: if classification then
Step 7: let each of the k models classify X and return the majority vote;
Step 8: if prediction then
Step 9: let each of the k models predict a value $f_i(1)$ initialize the weight of each tuple in D to $1=d$;
Step 10: for $i = 1$ to k do // for each round:
Step 11: sample D with replacement according to the tuple weights to obtain D_i ;
Step 12: use training set D_i to derive a model, M_i ;
Step 13: compute $\text{error}(M_i)$, the error rate of M_i
Step 14: if $\text{error}(M_i) > 0.5$ then
Step 15: reinitialize the weights to $1=d$
Step 16: go back to step 3 and try again;
Step 17: endif
Step 18: for each tuple in D_i that was correctly classified do
Step 20: multiply the weight of the tuple by $\text{error}(M_i) = (1 - \text{error}(M_i))$; // update weights
Step 21: normalize the weight of each tuple;
Step 22: endfor
Step 23: initialize weight of each class to 0;
Step 24: for $i = 1$ to k do // for each classifier:
Step 25: $w_i = \log 1 - \text{error}(M_i)$
Step 26: $\text{error}(M_i)$; // weight of the classifier's vote
Step 27: $c = M_i(X)$; // get class prediction for X from M_i
Step 28: add w_i to weight for class c
Step 29: endfor
Step 30: return the class with the largest weight; r X and return the average predicted value;

IV. RESULTS AND ANALYSIS FOR PREVIOUS TECHNIQUES

The result of research and at the same time is given the comprehensive discussion. Results can be presented in understand easily .

IG-ANFIS

IG is a feature selection technique used to reduce the number of input to ANFIS [3]. It uses ranking method and is often used in text categorization. The information gain is usually a not good measure for deciding the relevance of an attribute, it is not perfect. A notable problem occurs when information gain is applied to attributes that can take on a large number of distinct values. For example, suppose that one is building a decision tree for some data describing the customers of a business. Information gain is often used to decide which of the attributes are the most relevant, so they can be tested near the root of the tree [3]. One of the input attributes might be the customer's credit card number. This attribute has high mutual information, because it uniquely identifies each customer, but we do not want to include it in the decision tree this biases the decision tree against considering attributes with a large number of distinct values. However, attributes with very low information values then appeared to receive an unfair advantage.

GI-ANFIS

The Classification and Regression Trees (CART)[4], which described the generation of binary decision trees CART were invented independently of one another at around the same time, yet follow a similar approach for learning decision trees from training tuple. but one or more drawbacks GI-ANFIS are any index is that of over simplification. To begin with, absolute and relative levels of inequality are difficult to measure source.

GR-ANFIS

The split information of the attribute is too low; GR-ANFIS will try to split on the attribute[5]. A way out of this can be to set some threshold for the intrinsic information, that way we would be ignoring the problem causing attributes.

V. RESULTS AND ANALYSIS

```

18:21:13 - meta.Vote
=== Run information ===
Scheme:      weka.classifiers.meta.Vote -S 1 -B "weka.classifiers.meta.AdaBoostM1 -P 100 -S 1 -I 10 -W weka.classifiers.trees.DecisionStump" -B "weka.classifiers.meta.Bagging -P 100 -S 1 -I 10 -W weka.classifiers.trees.RepTree"
Relation:    data for ADBA
Instances:   331
Attributes:  15
  sl.no
  age
  sex
  occupation
  cp(4 value)
  trestbps(initial stage 3 types)
  cholesterol(more than < 4.5)
  fb(fasting blood sugar)
  restecg(resting electrocardiographic results )- three level
  thalach (heart rate less 100 or more than 100)
  exang
  oldpeak(ST depression induced by exercise relative to rest )
  slope( the slope of the peak exercise ST segment)
  ca( ca: number of major vessels (0-3) colored by flourosopy )
  foodhabits among heart diseases result
Test mode:  evaluate on training data

=== Classifier model (full training set) ===
Vote combines the probability distributions of these base learners:
weka.classifiers.meta.AdaBoostM1 -P 100 -S 1 -I 10 -W weka.classifiers.trees.DecisionStump
weka.classifiers.meta.Bagging -P 100 -S 1 -num-slots 1 -I 10 -W weka.classifiers.trees.RepTree -- -M 2 -V 0.001 -N 3 -S 1 -L -1 -I 0.0
using the 'Maximum' combination rule

All the models:
AdaBoostM1: Base classifiers and their weights:
Decision Stump
Classifications
  
```

Fig 3: To view of integrated AD-BA Algorithm with help of voting interface

```

18:21:13 - meta.Vote
Bagging with 10 iterations and base learner
weka.classifiers.trees.RepTree -M 2 -V 0.001 -N 3 -S 1 -L -1 -I 0.0

Time taken to build model: 0.05 seconds
=== Evaluation on training set ===
Time taken to test model on training data: 0 seconds
=== Summary ===
Correctly Classified Instances      328          99.0937 %
Incorrectly Classified Instances    3            0.9063 %
Kappa statistic                    0.9859
Mean absolute error                 0.2105
Root mean squared error             0.2477
Relative absolute error             65.3358 %
Root relative squared error        61.7867 %
Total Number of Instances          331

=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
      1.000    0.000    1.000     1.000    1.000     1.000    1.000    1.000    junk food
      0.985    0.004    0.985     0.985    0.985     0.981    0.999    0.998    highcholesterol food
      1.000    0.005    0.993     1.000    0.997     0.994    1.000    1.000    oil food
      0.983    0.005    0.991     0.983    0.987     0.980    0.999    0.999    junk food
Weighted Avg.   0.991    0.005    0.991     0.991    0.991     0.987    1.000    0.999

=== Confusion Matrix ===
 a  b  c  d  <-- classified as
 2  0  0  0  | a = junk food
 0  67 0  1  | b = highcholesterol food
 0  0 145 0  | c = oil food
 0  1  1 114  | d = junk food
  
```

Fig4: To view combine of AD-BA result

The dataset used for AD-BA Algorithm is shown in result. It is same as the original dataset are using more than 300 values and 14 attributes. The weka tool to support for combines AD-BA Algorithm to analysis and accuracy to envisage heart attack.

VI. COMPARISON FOR RESULT ACCURACY

THE APPROCH	ACCURACY
IG-ANFIS	97.24%
GI-ANFIS	98.56%
GR-ANFIS	98.62%
BOOTSTRAP(AD-BA)	99.09 %

Tab1: To view of Bootstrap (AD-BA) accuracy

This Results shown as various types approach to display, especially BOOTSTRAP approach to compare previous three types of approach to give the better result for enhancing the heart attack envisage.

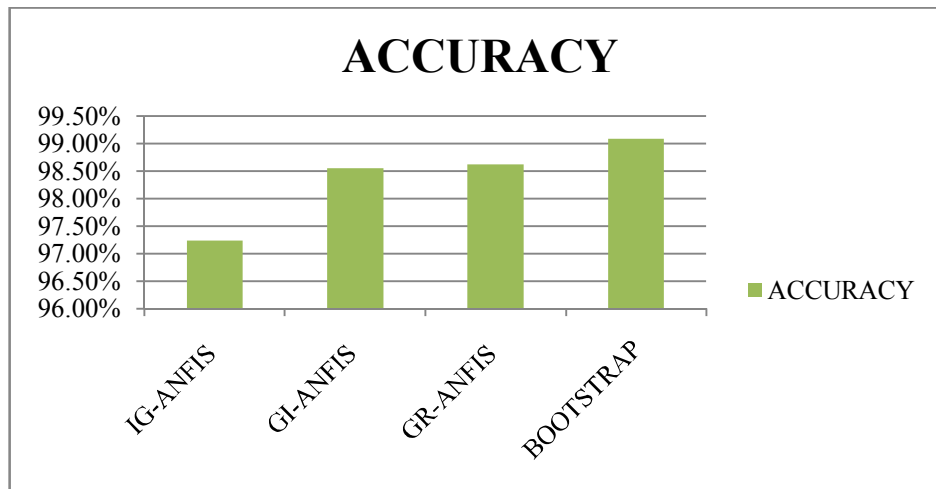


Fig5: To view of Bootstrap (AD-BA) accuracy

The accuracy maintains the greatest entitlement show to above the chart. The BOOTSTRAP approaches to weigh against previous approach to furnish very excellent performance at 99.06 %, better accuracy for heart attack envisages.

VII. CONCLUSION

In this dissertation, collection of knowledge classification and prediction models have been developed to analyse and classify the occurrence and absence of heart disease in patient result forecast moreover, the Bootstrap method. The developed classification and prediction models, based on the AD-BA algorithm, were collection learning classifiers that had combine in two algorithm for using voting process to spawn a well-built, single complex collection knowledge classification and prediction model by using an most favourable weighted popular choose of a number of puny classifiers.

REFERENCE

1. Diagnosing Coronary Heart Disease Using Ensemble Machine Learning Kathleen H. Miao, Julia H. Miao, and George J. Miao International Journal of Advanced Computer Science and Applications, Vol. 7, No. 10, 2016
2. Deepali Chandna, et al., "Diagnosis of Heart Disease Using Data Mining Algorithm, International Journal of Computer Science and Information Technologies", Vol. 5 (2), 2014.
3. Benard Nyangena Kiage, et al., "IG-ANFIS DATA MINING APPROACH FOR FORECASTING CANCER THREATS", International Journal of Engineering Research And Management (IJERM), Volume-01, Issue-06, September 2014.
4. S.S.Baskar, et al., "Filter Based Feature Selection Approach to Enhance the Classification Accuracy", 2013
5. W. Altidor, T. M. Khoshgoftaar, J. Van Hulse, A. Napolitano, et al., "Ensemble feature ranking methods for data intensive computing applications", Handbook of data intensive computing, Springer Science + Business media, LLC 2011, pages 349 -376.
6. Y. Saeys, T. Abeel, et al., Y. V. Peer, "Robust feature selection using ensemble feature selection techniques", W. Daelemans et al. (Eds.): ECML PKDD 2008, Part II, LNAI 5212, pages 313-32
7. A Emam, Ischemia prediction using ANFIS, "Institute of Electrical and Electronics Engineers", 2010.
8. J Healthc Eng, "Neural Network-Based Coronary Heart Disease Risk Prediction Using Feature Correlation Analysis", 2017
9. Yen GG, Meesad P, et al., "Constructing a fuzzy rule-based system using the ILFN network and Genetic Algorithm", 2001.