Improving the Performance of Data Mining Algorithms in Health Care Data

[1]P. Santhi, [2]V. Murali Bhaskaran,
[1,2]Paavai Engineering College, India

Abstract
The healthcare industry is one of the most important industries in the world. In this industry having the large amounts of healthcare data .This data is used for the many purposes. In this industry the heart disease is most challenging problem. The health care industry is having the many data related to the human health. In this paper proposes the performance of clustering and classification algorithm using heart disease data. We evaluating the performance of classifiers of Bayes (Naïve Bayes, Naïve Bayes updateable), functions (SMO), Lazy (IB1, IBK), Meta Multi BoostAB, Multiclass Classifier), Rule(Decision Table), trees(NB Tree) and the clustering algorithms of EM, Cobweb, Farthest First, Make Density Based Clusters, Simple K-Means algorithms. The performance of classification will be calculated by using the cross validation test mode and the performance of clusters will be calculated using the mode of classes to clusters evaluation. In the final result shows the high performance algorithms for heart disease data.

Keywords
Data mining, Classification, Clustering, NB Tree, K-means

I. Introduction

The healthcare industry is the main industries in the world. A health care provider is an institution or person that provides preventive, curative, promotional or rehabilitative health care services in a systematic way to individuals, families or community. In health care the data mining is more popular and essential for all the healthcare applications [3]. It contains the many data, but these data have not been used for some useful purpose. This data will be converted in to the some useful purpose by using data mining techniques. Heart disease is a general name for a wide variety of diseases, disorders and conditions that affect the heart and sometimes the blood vessels as well [4]. Heart disease is the number one killer of women and men. Symptoms of heart disease vary depending on the specific type of heart disease. A classic symptom of heart disease is chest pain. However, with some forms of heart disease, such as atherosclerosis, there may be no symptoms in some people until life-threatening complications develop. Any of a number of conditions that can be affects the heart. The data mining is the process of finding the hidden knowledge from the data base or any other information repositories. The main purpose of the health care industry is to improving the quality of healthcare data by reducing the missing values and removing the noise in the data base. In existing papers, having three algorithms for predicting the heart disease. In this paper proposes the more than three algorithms for finding the performance of classifiers and clusters based on the heart disease data.

II. Evaluation steps

In this paper having the following evaluation steps for finding the performance of classification and the clustering.

1. Dataset Collection
2. Data Preprocessing
3. Classification
4. Clustering

A. Dataset Collection

This data will be collected from the Switzerland data base. The data set having the attributes of age, sex, Chest Pain, Blood pressure, Blood sugar, etc. In this paper having the 107 instances. And 14 attributes. These attributes are mainly used for predicting the heart disease and calculating the performance of these algorithms.

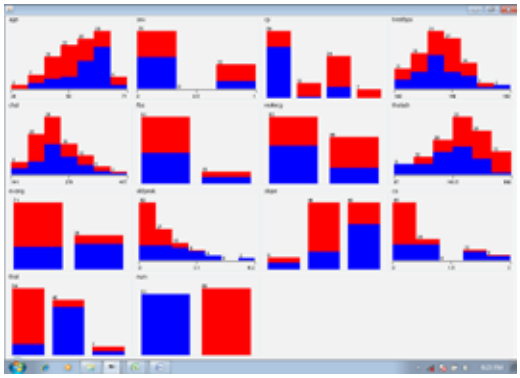

Fig.1 : Visualization of Attributes before the Preprocessing

B. Data preprocessing

Data pre-processing is an often neglected but important step in the data mining process[3]. The phrase "Garbage In, Garbage Out" is particularly applicable to data mining and machine learning projects. Data gathering methods are often loosely controlled, resulting in out-of-range values, impossible data combinations, missing values, etc. The data preprocessing having the algorithms of Data Cleaning ,Data Transformation, Data Reduction, Data Integration and Normalization [1]. The Data Cleaning is the process of removing noisy data, inconsistent and redundant data. In Data transformation the data will be converted in to the data mining process. The Data integration is the process of combining the data bases in to the data ware house. After the preprocessing, the data will be given to the data mining process. In this paper introduces the attribute subset selection algorithm for data preprocessing.

1. Attribute Subset Selection

In weka the preprocessing contains two filters of supervised and unsupervised filters. The attribute selection is one of the supervised filters. A supervised attribute filter that can be used to select attributes [8]. It is very flexible and allows various search and evaluation methods to be combined. In this filter uses the CfsSubsetEval for the evation and the best first for the searching.

Options

evaluator -- Determines how attributes/attribute subsets are evaluated. search -- Determines the search method.

(i). CfsSubsetEval

Evaluates the worth of a subset of attributes by considering the individual predictive ability of each feature along with the degree of redundancy between them. Subsets of features that are highly correlated with the class while having low intercorrelation are preferred.

Options

locallyPredictive -- Identify locally predictive attributes. Iteratively adds attributes with the highest correlation with the class as long as there is not already an attribute in the subset that has a higher correlation with the attribute in question missingSeparate -- Treat missing as a separate value. Otherwise, counts for missing values are distributed across other values in proportion to their frequency.

(ii). Best First

Searches the space of attribute subsets by greedy hill climbing augmented with a backtracking facility. Setting the number of consecutive non-improving nodes allowed controls the level of backtracking done. Best first may start with the empty set of attributes and search forward, or start with the full set of attributes and search backward, or start at any point and search in both directions (by considering all possible single attribute additions and deletions at a given point).

Options

direction -- Set the direction of the search. lookupCacheSize -- Set the maximum size of the lookup cache of evaluated subsets. This is expressed as a multiplier of the number of attributes in the data set. (default = 1). searchTermination -- Set the amount of backtracking. Specify the number of startSet -- Set the start point for the search. This is specified as a comma seperated list off attribute indexes starting at 1.

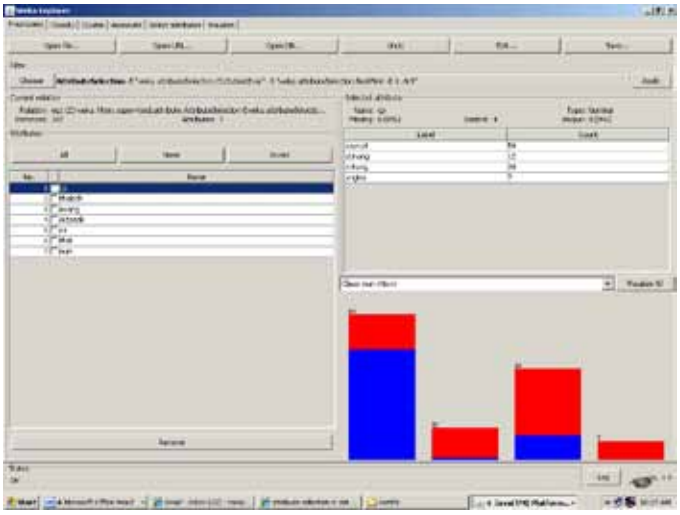

Fig.2 : Attribute Selection

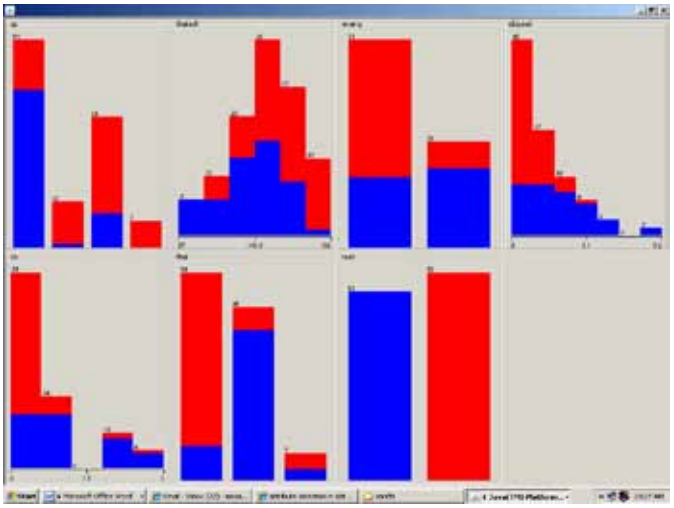

Fig.3 :Visualization of attributes after preprocessing

C. Classification

The classification is the method of supervised learning [6]. It is the task of generalizing known structure to apply to new data [6]. In this classification contains the classifiers of Bayes, functions, Lazy, Meta, Misc and Tree classifiers [7].

1. Building Bayes Classifiers Algorithm

memory problems may occur. Switching this option off makes the structure learning algorithms slower, and run with less memory. By default, ADTrees are used.

(i). Naïve Bayes

The Class is used for a Naive Bayes classifier using estimator classes [1]. Numeric estimator precision values are chosen based on analysis of the training data. For this reason, the classifier is not an UpdateableClassifier (which in typical usage are initialized with zero training instances) -- if you need the Updateable Classifier functionality, use the NaiveBayes Updateable classifier [8]. The Naive Bayes Updateable classifier will use a default precision of 0.1 for numeric attributes when build Classifier is called with zero training instances.

Options

Debug -- If set to true, classifier may output additional info to the console. Use Kernel Estimator -- Use a kernel estimator for numeric attributes rather than a normal distribution. Use Supervised Discretization -- Use supervised discretization to convert numeric attributes to nominal ones.

(ii). Naïve Bayes Uptateable

The Class is used for a Naive Bayes classifier using estimator classes. This is the updateable version of Naïve Bayes. This classifier will use a default precision of 0.1 for numeric attributes when build Classifier is called with zero training instances.

Options

Debug -- If set to true, classifier may output additional info to the console. Use KernelEstimator -- Use a kernel estimator for numeric attributes rather than a normal distribution. Use SupervisedDiscretization -- Use supervised discretization to convert numeric attributes to nominal ones.

2. Building Function Classifiers Algorithm

(i). SMO

It Implements the John Platt's sequential minimal optimization algorithm for training a support vector classifier. This implementation globally replaces all missing values and transforms nominal attributes into binary ones. It also normalizes all attributes by default. (In that case the coefficients in the output are based on the normalized data, not the original data --- this is important for interpreting the classifier.) Multi-class problems are solved using pair wise classification. To obtain proper probability estimates, use the option that fits logistic regression models to the outputs of the support vector machine. In the multi-class case the predicted probabilities are coupled using Hastie and Tibshirani's pairwise coupling method.

Options

Build LogisticModels -- Whether to fit logistic models to the outputs (for proper probability estimates). c -- The complexity parameter C. Cache Size -- The size of the kernel cache (should be a prime number). Use 0 for full cache. Debug -- If set to

true, classifier may output additional info to the console. Epsilon -- The epsilon for round-off error (shouldn't be changed). Exponent -- The exponent for the polynomial kernel. Feature Space Normalization -- Whether feature-space normalization is performed (only available for non-linear polynomial kernels). Filter Type -- Determines how/if the data will be transformed. Gamma -- The value of the gamma parameter for RBF kernels. Lower Order Terms -- Whether lower order polyomials are also used (only available for non-linear polynomial kernels). Num Folds -- The number of folds for cross-validation used to generate training data for logistic models (-1 means use training data). Random Seed -- Random number seed for the cross-validation. Tolerance Parameter -- The tolerance parameter (shouldn't be changed).

3. Building Lazy Classifiers Algorithm

(i). IB1

Nearest-neighbour classifier. Uses normalized Euclidean distance to find the training instance closest to the given test instance, and predicts the same class as this training instance. If multiple instances have the same (smallest) distance to the test instance, the first one found is used.

Options

Debug -- If set to true, classifier may output additional info to the console.

(ii). IBK

Storing and using specific instances improves the performance of several supervised learning algorithms. These include algorithms that learn decision trees, classification rules, and distributed networks. However, no investigation has analyzed algorithms that use only specific instances to solve incremental learning tasks [8]. In this paper, we describe a framework and methodology, called instance-based learning that generates classification predictions using only specific instances. Instance-based learning algorithms do not maintain a set of abstractions derived from specific instances. This approach extends the nearest neighbor algorithm, which has large storage requirements. We describe how storage requirements can be significantly reduced with, at most, minor sacrifices in learning rate and classification accuracy. While the storage-reducing algorithm performs well on several real-world databases, its performance degrades rapidly with the level of attribute noise in training instances. Therefore, we extended it with a significance test to distinguish noisy instances. This extended algorithm's performance degrades gracefully with increasing noise levels and compares favorably with a noise-tolerant decision tree algorithm.

4. Building Meta Classifiers Algorithm

(i). MultiBoostAB

This Class is used for boosting a classifier using the Multi Boosting method. Multi Boosting is an extension to the highly successful AdaBoost technique for forming decision committees. Multi Boosting can be viewed as combining AdaBoost with wagging. It is able to harness both Ada Boost's high bias and variance reduction with wagging's superior variance reduction. Using C4.5 as the base learning algorithm, Multi-boosting is demonstrated to produce decision committees with lower error than either AdaBoost or wagging significantly more often than the reverse over a large representative cross-section of UCI data sets. It offers the further advantage over AdaBoost of suiting parallel execution.

(ii). Multi class classifier

A meta classifier is used for handling multi-class datasets with 2-class classifiers. This classifier is also capable of applying error correcting output codes for increased accuracy.

Options

Classifier -- The base classifier to be used. Debug -- If set to true, classifier may output additional info to the console. Method -- Sets the method to use for transforming the multi-class problem into several 2-class ones. Random Width Factor -- Sets the width multiplier when using random codes. The number of codes generated will be thus number multiplied by the number of classes. Seed -- The random number seed to be used.

5. Building Rule Classifiers

(i). Decision Table

The Class for building and using a simple decision table majority classifier. It evaluates feature subsets using best-first search and can use cross-validation for evaluation. There is a set of methods that can be used in the search phase (E.g.: Best First, Rank Search, Genetic Search) and we may also use LBk to assist the result. In this experiment, we Choose the cross Val = 1; search Method = Best First and useIBk = False

6. Building Tree Classifiers Algorithm

(ii). NB Tree

The Class is for generating a decision tree with naive Bayes classifiers at the leaves.

Options

Debug -- If set to true, classifier may output additional info to the console.

Table 1: Prediction Accuracy of Classifiers

Classifiers Category	Classifiers Algorithm	Prediction Accuracy
Bayes	Naïve Bayes, Naïve Bayes Updateable	88.8%
Function	SMO	86%
Lazy	IB1,IBk	83.6%
Meta	MultiBoostAB, Multi Class Classifier	86.1%
Rule	Decision Table	88.8%
Tree	NB Tree	90.7%

D. Clustering

The Clustering is the process of grouping the similar data items [2]. It is the unsupervised learning techniques, in which the class label will not be provided. The Clustering methods are Partitioned clustering, Hierarchical methods, Density based clustering, Sub Space Clustering. Hierarchical algorithms find successive clusters using previously established clusters. These algorithms usually are either agglomerative ("bottom-up") or divisive ("top-down"). Agglomerative algorithms begin with each element as a separate cluster and merge them into successively larger clusters. Divisive algorithms begin with the whole set and proceed to divide it into successively smaller clusters. Partitioned algorithms typically

determine all clusters at once, but can also be used as divisive algorithms in the hierarchical clustering [9]. Density-based clustering algorithms are devised to discover arbitrary-shaped clusters. In this approach, a cluster is regarded as a region in which the density of data objects exceeds a threshold. DBSCAN and OPTICS are two typical algorithms of this kind. Subspace clustering methods look for clusters that can only be seen in a particular projection (subspace, manifold) of the data. These methods thus can ignore irrelevant attributes. The general problem is also known as Correlation clustering while the special case of axis-parallel subspaces is also known as Two-way clustering, co-clustering or bi clustering: in these methods not only the objects are clustered but also the features of the objects, i.e., if the data is represented in a data matrix, the rows and columns are clustered simultaneously [1]. They usually do not however work with arbitrary feature combinations as in general subspace methods. But this special case deserves attention due to its applications in bioinformatics. Conceptual clustering is a machine learning paradigm for unsupervised classification developed mainly during the 1980s. It is distinguished from ordinary data clustering by generating a concept description for each generated class [9]. Most conceptual clustering methods are capable of generating hierarchical category structures; see Categorization for more information on hierarchy. Conceptual clustering is closely related to formal concept analysis, decision tree learning, and mixture model learning. The clustering having the measures of Manhatten, Euclidean, Minkowski and Hamming Distance.

1. Building of Clustering Algorithm

The clustering algorithms will be building by using the heart disease data [10]. The algorithms are Cob web, EM, Make Density Based Clusters and Simple K-Means.

(i). Expectation Maximization

It is the method in partitioned clustering. The expectation-maximization algorithm is a more statistically formalized method which includes some of these ideas: partial membership in classes [7]. It generally preferred to fuzzy-c-means. The EM algorithm can also accommodate categorical variables. The method will at first randomly assign different probabilities (weights, to be precise) to each class or category, for each cluster. In successive iterations, these probabilities are refined (adjusted) to maximize the likelihood of the data given the specified number of clusters. The results of EM clustering are different from those computed by k-means clustering. The latter will assign observations to clusters to maximize the distances between clusters. The EM algorithm does not compute actual assignments of observations to clusters, but classification probabilities. In other words, each observation belongs to each cluster with a certain probability. Of course, as a final result you can usually review an actual assignment of observations to clusters, based on the (largest) classification probability. The EM (expectation maximization) algorithm extends this basic approach to clustering in two important ways: Instead of assigning cases or observations to clusters to maximize the differences in means for continuous variables, the EM clustering algorithm computes probabilities of cluster memberships based on one or more probability distributions. The goal of the clustering algorithm then is to maximize the overall probability or likelihood of the data, given the (final) clusters. Unlike the classic implementation of k-means clustering, the general EM algorithm can be applied to both continuous and categorical variables (note that the classic k-means algorithm can also be modified to accommodate categorical variables).

Options

maxIterations -- maximum number of iterations
minStdDev -- set minimum allowable standard deviation
numClusters -- set number of clusters. -1 to select number of clusters automatically by cross validation.
seed -- random number seed

(ii). COBWEB

COBWEB is an incremental system for hierarchical conceptual clustering. COBWEB incrementally organizes observations into a classification tree. Each node in a classification tree represents a class (concept) and is labeled by a probabilistic concept that summarizes the attribute-value distributions of objects classified under the node. This classification tree can be used to predict missing attributes or the class of a new object. There are four basic operations COBWEB employs in building the classification tree. Which operation is selected depends on the category utility of the classification achieved by applying it. The operations are:

Merging Two Nodes

IT merging two nodes means replacing them by a node whose children is the union of the original nodes' sets of children and which summarizes the attribute-value distributions of all objects classified under them.

Splitting a node

A node is split by replacing it with its children.

Inserting a new node

A node is created corresponding to the object being inserted into the tree.

Passing an object down the hierarchy

IT effectively calling the COBWEB algorithm on the object and the sub tree rooted in the node.

(iii). Farthest First Algorithm

Farthest first is a Variant off K means that places each cluster centre in turn at the point furthest from the existing cluster centers. This point must lie within the data area. This greatly sped up the clustering in most cases since less reassignment and adjustment is needed.

Options

numClusters -- set number of clusters
seed -- random number seed

(iv). Make Density Based Clusters

The cluster will be constructed based on the density properties of the database are derived from a human natural clustering approach. The clusters and consequently the classes are easily and readily identifiable because they have an increased density with respect to the points they possess. The elements of the database can be classified in two different types: the border points, the points located on the extremities of the cluster, and the core points, which are located on its inner region.

(v). Simple K-Means

It is based on the partitioned clustering. The k-means algorithm assigns each point to the cluster whose

Table 2: Prediction Accuracy of Clusters

Cluster Category	Clusters Algorithms	Measures		
		Correctly Classified Instance	In correctly Classified Instance	Prediction Accuracy
Clusters	COBWEB	3	104	2.8
	EM	92	15	85.98
	Farthest First	72	35	67..29
	Make Density Based Clusters	94	13	87.85
	Simple K-Means	89	18	83.18

center (also called centroid) is nearest. The center is the average of all the points in the cluster — that is, its coordinates are the arithmetic mean for each dimension separately over all the points in the cluster.

Options

numClusters -- set number of clusters seed -- random number seed

III. Performance evaluation

The above table shows the performance of clustering and classification algorithms using heart disease data. The attributes will be evaluated based on the prediction accuracy of the algorithms. The following evaluation graph shows the performance of classifiers and the clustering algorithms.

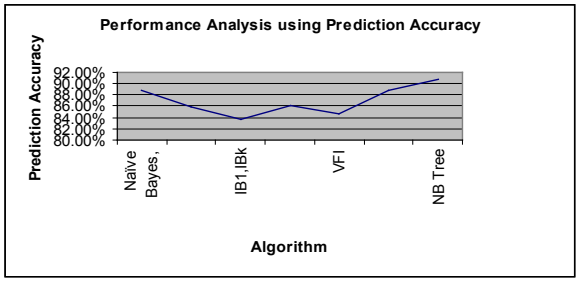

Fig.4 : Evaluation graph for Classifiers using Prediction Accuracy

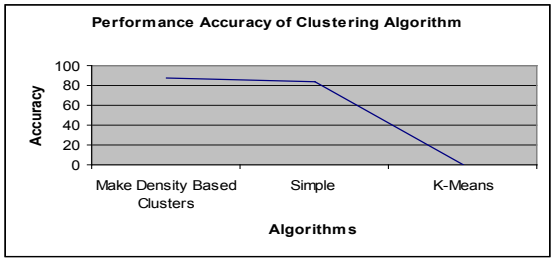

Fig.5 : Evaluation graph for Clusters using Prediction Accuracy

In data mining the classification algorithms NB tree having the highest prediction accuracy comparing to clustering algorithm. In classification the NB tree has 90.7% of accuracy and the clustering Simple K-means having the 83.18% of Prediction Accuracy. The following visualization shows the error and cost curve for sick and buff classes in the classifiers and the simple k-means for the clustering algorithm.

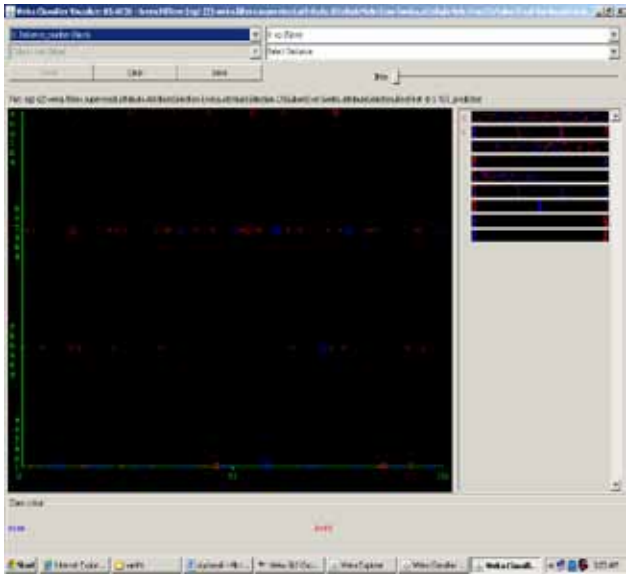

Fig.6 : Visualization of NB Tree classifier

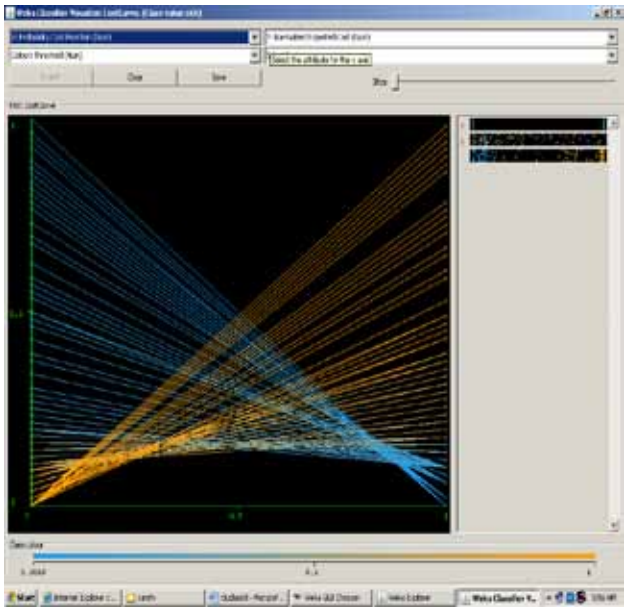

Fig.7 : Cost Curve for NB tree for Sick Classes

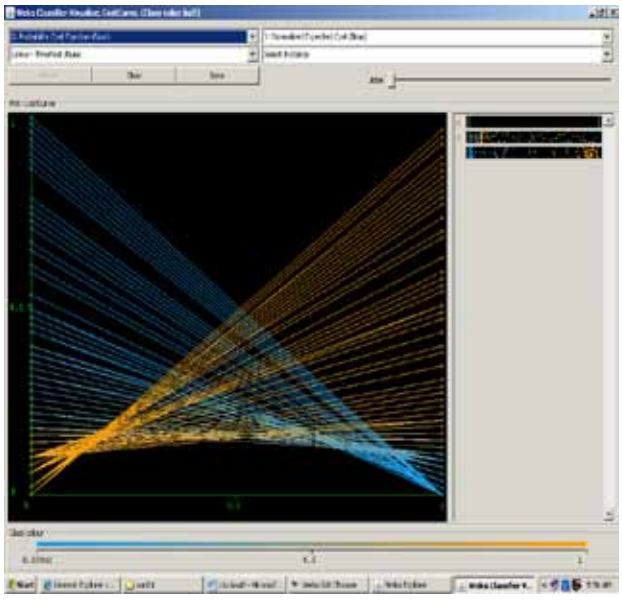

Fig.8 : Cost Curve for NB tree for buff Classes

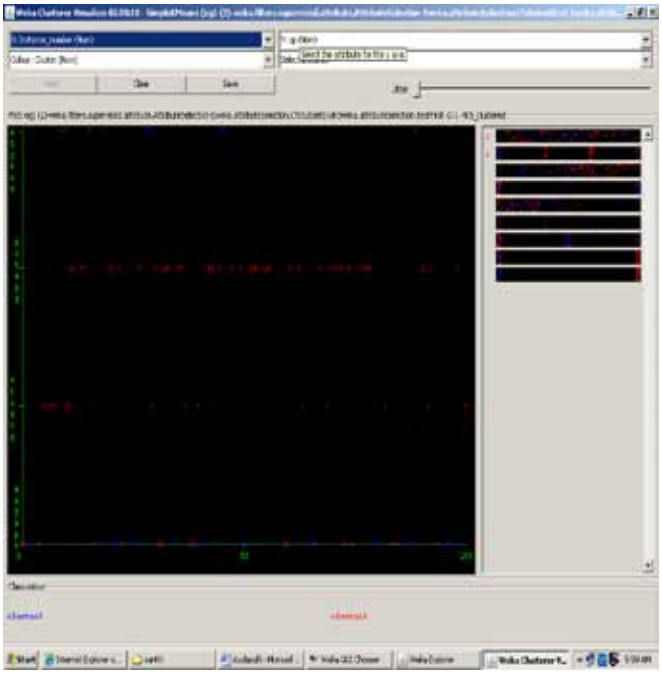

Fig.9 : Visualization for Simple K-Means Algorithm

IV. Conclusion

In healthcare industry having the large amount of useful data. In this data is used for many purposes, here the heart attack prediction data is used for find the performance of classifiers and the clustering algorithm. In final result shows the performance of classifiers and the cluster algorithms using prediction accuracy. In this result shows the cost curve of sick and buff classes. The comparison result shows that, the NB Tree having the highest prediction Accuracy comparing to the clustering algorithm.

References

[1] Varun Kumar, Nisha Rathee,"Knowledge Discovery from Database using an Integration of clustering and Classification", IJACSA, vol 2 No.3,PP. 29-33,March 2011.

[2] Ritu Chauhan, Harleen Kaur, M.Afshar Alam, "Data Clustering Method for Discovering Clusters in Spatial Cancer Databases", International Journal of Computer Applications (0975 – 8887) Volume 10– No.6, November 2010.

[3] G.Karraz,G.Magenes,"Automatic Classification of Heart beats using Neural Network Classifier based on a Bayesian Frame Work", IEEE, Vol 1,2006.

[4] N.A.Setiawan, A.F.M.Hani,"Missing Attribute Value Prediction Based on Artificial Neural Network and Rough Set Theory", IEEE, Vol 1, pp.306-310,2008.

[5] Sellappan Pandian, Rafigh Awang,"Heart Disease Prediction System using Data Mining Techniques",IEEE Computer, Vol 7, PP.295-304, August 2008.

[6] Shantakumar B.Patil, Dr.Y.S.Kumaraswamy,"Extraction of Significant Patterns from Heart Disease Warehouses for Heart Attack Prediction", IJCSNS International Journal of Computer Science and Network Security, VOL.9 No.2, February 2009.

[7] K.Srinivas, B.Kavihta Rani, Dr. A.Govrdhan,"Applications of Data Mining Techniques in Healthcare and Prediction of Heart Attacks", International Journal on Computer Science and Engineering,Vol. 02, No. 02, 250-255,2010.

[8] Weka, "Data Mining Machine Learning Software, [Online] Available : http://www.cs.waikato.ac.nz/ml/.

[9] Witten, I.H., Frank, E, "Data Mining: Practical Machine Learning Tools and Techniques", 2nd edn. Morgan Kaufmann, San Francisco (2005).

[10] Ian H.Witten, et al, "Weka: Practical Machine Learning Tools and Techniques with Java implementations," Working Paper 99/11, Department of Computer Science, The University of Waikato, Hamilton, 1999.