Analysis of cancer datasets using Classification Algorithms

Parvesh Kumar
Department of Computer Science
Maharaja Surajmal Institute, Delhi(India)

Siri Krishan Wasan
Department of Mathematics
Jamia Millia Islamia, Delhi(India)

Abstract: Cancer detection is one of the important research topics in medical science. In bioinformatics age, gene expression data can be used for the cancer detection. Data mining techniques, such as pattern association, classification and clustering, are now frequently applied in cancer and gene expressions correlation studies. Classification is very important among these techniques of data mining. Here in this paper we studied various classification algorithms like C4.5, CART, Random Forest, LMT, ADT, Naïve Bayesian and Bayesian logistic Regression over different cancer dataset. Accuracy is the main objective to estimate the performance of these algorithms over cancer datasets.

Keywords: Data mining, Classification, Decision tree

Introduction:

Classification of data objects based on a predefined knowledge of the objects is a data mining and knowledge management technique used in grouping similar data objects together. It can be defined as supervised learning algorithms as it assigns class labels to data objects based on the relationship between the data items with a pre-defined class label. Classification algorithms have a wide range of applications like churn prediction, fraud detection, artificial intelligence, and credit card rating etc. there are many classification algorithms available in literature. Classification is a well-studied area in data mining. Numerous classification algorithms have been proposed in the literature, such as decision tree classifiers (Quinlan, 1993), rule-based classifiers(Cohen,1995), Bayesian classifiers(Langley et al, 1992), support vector machines(SVM) (Vapnik, 1995), artificial neural networks(Andrews et al., 1995), Lazy Learners, and ensemble methods(Dietterich), 2000.

To classify the various types of cancer into its different subcategories, different data mining techniques have been used over gene expression data. One might want to partition the data set to find naturally occurring groups of genes with similar expression patterns. Golub (Golub et al, 1999), Alizadeh (Alizadeh et al., 2000) and Nielsen(Nielsen et al., 2002) have considered the classification of cancer types using gene expression datasets. A new classification method *(*Fort et al., 2005*)* is proposed combining partial least squares (PLS) and Ridge penalized logistic regression. This procedure is compared with other Classifiers using predictive performance of the resulting classification rules on three cancer data sets: Leukemia, Colon and Prostate. A study of Cancer Surveillance using Data Warehousing, Data Mining, and Decision Support Systems (Forgionne et al., 2000) discussed how data warehousing, data mining, and decision support systems can reduce the national cancer burden or the oral complications of cancer therapies.

Here in this paper , we studied various classification algorithms like C4.5, CART, Random Forest, LMT, ADT, Naïve Bayesian and Bayesian logistic Regression over different cancer dataset. In first few sections we briefly described these algorithms and after that an overview of cancer datasets is given. Results are discussed in the last section of the paper.

C4.5 :

C4.5(Quinlan,1993) is an algorithm used to generate a decision tree for classification developed by Ross Quilan. C4.5 is basically an extension of ID3 algorithm that accounts for missing values, continuous attribute value ranges and pruning of decision tree. C4.5 builds decision trees from a set of training data in the same way as ID3, using the concept of gain ratio based on information entropy. At each node of the tree, C4.5 chooses one attribute of the data that have maximum gain ratio to splits its set of samples into subsets enriched in one class or the other. This algorithm has a few base cases:-

- All the samples in the list belong to the same class. When this happens, it simply creates a leaf node for the decision tree saying to choose that class.
- None of the features provide any information gain. In this case, C4.5 creates a decision node higher up the tree using the expected value of the class.

Manuscript received June 5, 2010
Manuscript revised June 20, 2010

- Instance of previously-unseen class encountered. Again, C4.5 creates a decision node higher up the tree using the expected value.

In general, steps in C4.5 algorithm to build a decision tree are given below:

- Step1: For each attribute compute gain ratio using the formula given below and select one with maximum gain ratio.
- Step2: Create branch for each value of that attribute if attribute is discrete otherwise divide in range.
- Step3: Split cases using splitting criterion until subset contains one class samples.
- Step4: Repeat process for each branch until all cases in the branch have the same class.

Splitting Criteron:

The splitting criterion used in C4.5 is information gain ratio. The formula for information gain ratio of an attribute X for a set of cases T={T_1, T_2, T_3,….T_s} is calculated as follows:

$$\boldsymbol{GainRatio(X,T)} = \frac{\boldsymbol{Gain(X,T)}}{\boldsymbol{info(X,T)}}$$

Where $\boldsymbol{gain = info(T) - \sum_1^s \frac{|T_i|}{|T|} \times info(T_i)}$

and $\boldsymbol{info(T) = -\sum_{j=1}^{NClass} \frac{freq(C_j,T)}{|T|} \times \log_2\left(\frac{freq(C_j,T)}{|T|}\right)}$ is the formula for entropy.

Pruning in C4.5 :

Two techniques of pruning are implemented in C4.5 – first with subtree replacement by a leaf node if the error rate is close to error rate of original tree and replacement is worked from bottom to the root. In Second pruning technique, a subtree is replaced by its most used subtree and subtree is raised from its current location to a node higher up in the tree if error rate is not significant.

CART:

CART algorithm (Breiman et al., 1984) is a data mining algorithm, which is widely used statistical procedure based on tree structure that can produce classification and regression trees, depending on whether the dependent variable is categorical or numeric, respectively and generates binary tree.

CART is a recursive and gradual refinement algorithm of building a decision tree. To predict the classification situation of new samples of known input variable value, we only need to trace back downwards the decision tree model, compare the threshold value of new sample and the node variable at every node, and select appropriate branches until leaf nodes are reached.

Trees are formed by a collection of rules based on values of certain variables in the modeling data set.Rules are selected based on how well splits based on variables' values can differentiate observations based on the dependent variable

- Step1: All rows in a dataset are assigned to the root node.
- Step2: Each of the predictor variables is split at all its possible split points based on their values for the rows in the node considered.
- Step3: For each split point, the parent node is split into two child nodes by separating the rows with values lower than or equal to the split point and values higher than the split point for the considered predictor variable. For categorical predictor variables, each category of the variable will be considered in turn.
- Step4: The predictor variable and split point with the highest value of I(formula is given below) is selected for the node.

  $$\boldsymbol{I(s/t) = 2P_L P_R \sum_{j=1}^{m} \left|P(C_j|t_L) - P(C_j|t_R)\right|}$$

  where P_L & P_R are the probabilities of a sample to lie in left sub-tree & right sub-tree respectively and $P(C_j|t_L)$ or $P(C_j|t_R)$ are the probabilities that a sample is in the class C_j and in the left sub-tree or right sub-tree.
- Step5: The split of the parent node into the two child nodes is performed based on the selected split point.
- Step6: Steps (2) to (5) are repeated, using each node as a new parent node, until the tree has the maximum size.
- Step7: The tree is pruned to select the optimal size tree.

Logistic Model Trees(LMT):

A logistic model tree (LMT)(Frank et al., 2005) is an algorithm for supervised learning tasks which is combined with linear logistic regression and tree induction. LMT creates a model tree with a standard decision tree structure with logistic regression functions at leaf nodes. In LMT , leaves have a associated logic regression functions instead of just class labels. The steps used in LMT algorithm are given below:

- Step1:(Growing Initial Tree) In this step, Initial linear regression model is built for root node using LogitBoost algorithm for whole dataset. Here LogitBoost is run on the dataset for a fixed number of iterations .

- Step2:(Splitting and stopping) Splitting criterion used in LMT algorithm is same as that used in C4.5 algorithm. After splitting the dataset , logistic regression models are then built at the child nodes on the corresponding subsets of dataset using LogicBoost algorithm. However initial weights and probability estimates are taken from the parent node. And splitting and model building continues until atleast 15 samples are present at node and a useful split is found.
- Step3:(Tree pruning) The CART algorithm is used for pruning of tree. CART pruning method uses a combination of training error and penalty term for model complexity to make pruning decisions.

Random Forest:

Random forest(Leo Breiman, 2001) is an ensemble classifier that consists of many decision tree and outputs the class that is the mode of the class's output by individual trees. The algorithm for inducing a random forest was developed by Leo Breiman and Adele Cutler. Random Forests grows many classification trees without pruning. Then a test sample is classified by each decision tree and random forest assigns a class which have maximum occurrence among these classifications. Each tree is constructed as follows:

- Step1: Let the number of training cases be N, and the number of variables in the classifier be M. Choose m input variables to be used to determine the decision at a node of the tree; m should be much less than M.
- Step2:Choose a training set for this tree by choosing N times with replacement from all N available training cases (i.e. take a bootstrap sample). Use the rest of the cases to estimate the error of the tree, by predicting their classes.
- Step3: For each node of the tree, randomly choose m variables on which to base the decision at that node. Calculate the best split based on these m variables in the training set. The value of m remains constant during forest growing. Random forest are sensitive to the value of m.
- Step4: Each tree is grown to the largest extent possible and not pruned as done in constructing a normal tree classifier.

Alternating decision tree:

An Alternating Decision Tree (ADTree) (Freund and Mason, 1999) is a machine learning method for classification. The ADTree data structure and algorithm are a generalization of decision tree and have connections to boosting. ADTrees were introduced by Freund and Mason. An alternating decision tree consists of decision nodes and prediction nodes. Decision nodes specify a predicate condition. Prediction nodes contain a single number. ADTrees always have prediction nodes as both root and leaves. An instance is classified by an ADTree by following all paths for which all decision nodes are true and summing any prediction nodes that are traversed. This is different from binary classification trees such as CART or C4.5 in which an instance follows only one path through the tree.

Description of the algorithm is given below:

The training set X contains m samples and each sample is of the form (x_i,y_i) where x_i is set of attributes & $\boldsymbol{x_i} \in \boldsymbol{R^d}$ and y_i is class variable & $\boldsymbol{y_i} \in \{\boldsymbol{-1}, \boldsymbol{+1}\}$.

- Step1: (Initialization) Set initial weights for each sample equal to 1(i.e. $w_{i,0}= 1$). Also set first rule R_1 to have precondition and condition which are both true. Assign predication value for this rule as $\boldsymbol{a = \frac{1}{2} ln \frac{w_{+(true)}}{w_{-(true)}}}$ where $\boldsymbol{w_{+(true)}}, \boldsymbol{w_{-(true)}}$ are total weights of the positive and negative samples for which condition is true. The initial precondition set is P_1={True}
- Step2: Change initial weights by new weights using $\boldsymbol{w_{i,1} = w_{i,0} e^{-a y_t}}$ where value for y_t is+1 or -1 for two class problems.
- Step3: Repeat for t=1 to T,
  1. Generate a set C of weak hypothesis using weights $w_{i,t}$.
  2. For each precondition $c_1 \in P_t$ and each condition $c_2 \in C$,get the values of c_1 and c_2 that minimizes the value of Z_t where
  $$\boldsymbol{Z_t = 2\left(\sqrt{W_+(c_1 \wedge c_2) W_-(c_1 \wedge c_2)} - \sqrt{W_+(c_1 \wedge\neg c_2) W_-(c_1 \wedge\neg c_2)}\right) + W(\neg c_1)}$$
  Where $\boldsymbol{W(\neg c_1) = W_+(\neg c_1) + W_-(\neg c_1)}$
  3. Set a new rule r_t with precondition c_1, condition c_2 and weights a and b given by $\boldsymbol{a = \frac{1}{2} ln \frac{w_+(c_1 \wedge c_2) + \varepsilon}{W_-(c_1 \wedge c_2) + \varepsilon}}$, $\boldsymbol{b = \frac{1}{2} ln \frac{w_+(c_1 \wedge \neg c_2) + \varepsilon}{W_-(c_1 \wedge \neg c_2) + \varepsilon}}$
  4. Set P_{t+1} by P_t with the addition of $c_1 \wedge c_2$ and $\boldsymbol{c_1 \wedge\neg c_2}$ and also set R_{t+1} by R_t with addition of new rule r_t.
  5. Update weights using $\boldsymbol{w_{i,t+1} = w_{i,t} e^{-r_t(x_i) y_t}}$
- Step4: (Output) Classification of test sample x is the sign of the sum of all the base rules in R_{T+1} i.e. $\boldsymbol{class(x) = sign\left(\sum_{t=1}^{T} r_t (x)\right)}$.

Naive Bayesian Classifier :

Naïve Bayesian classifier (Langley, 1995) based on Bayes conditional probability rule is used for performing classification tasks. Naive Bayes assumes the attributes are statistically independent which makes it an effective classification tool that is easy to interpret. It is best employed when faced with the problem of 'curse of dimensionality' i.e. when the number of attribute is very high.

A naive Bayes classifier is a simple probabilistic classifier based upon Bayes theorem with strong (naive) independence assumptions. All attributes of the dataset are considered independent of each other. In simple terms, a naive Bayes classifier assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature.

An advantage of the naive Bayes classifier is that it requires a small amount of training data to estimate the parameters (means and variances of the variables) necessary for classification. Because independent variables are assumed, only the variances of the variables for each class need to be determined and not the entire covariance matrix. The steps used in naïve bayesian algorithm are given below:

- Step1: calculate probability $P(C=C_j)$ for each class in the dataset.
- Step2: For each value x_i of each attribute a_i ,calculate probability $P(X_i|C=C_j)$
- Step3: Cassify new sample to class C_j that have maximum probability $\boldsymbol{P(C = C_j|X_1 \ldots . X_n)}$ using Naïve-Bayes Classifier given below

$$\boldsymbol{P(C = C_j|X_1 \ldots . X_n) = \frac{P(C = C_j)\prod_i P(X_i|C = C_j)}{\sum_j P(C = C_j)\prod_i P(X_i|C = C_j)}}$$

Or

$$\boldsymbol{C \leftarrow \arg max_{C_j} P(C = C_j)\prod_i P(X_i|C = C_j)}$$

because denominator does not depend upon the value of C_j.

BayesianLogisticRegression:

Logistic regression model is used for prediction of the probability of occurrence of an event by fitting data to a logistic curve. Logistic Regression is an approach to learning functions of the form $f : X \rightarrow C$, or $\boldsymbol{P(C = C_j|X)}$ in the case where C is discrete-valued, and $X = (X_1, X_2 \ldots\ldots.. X_n)$ is any vector containing discrete or continuous variables. Logistic Regression assumes a parametric form for the distribution $\boldsymbol{P(C = C_j|X)}$, then directly estimates its parameters from the training data. Logistic Regression directly estimates the parameters of $\boldsymbol{P(C = C_j|X)}$, whereas Naive Bayes directly estimates parameters for $\boldsymbol{P(C = C_j)}$ and **P(X$_i$|C=C$_j$).** (Genkin et al., 2004).

The parametric model assumed by Logistic Regression, when C is Boolean in nature, is given by

$$\boldsymbol{p(C = 1|X) = \frac{1}{1 + exp(w_0 + \sum_{i=1}^{n} w_i X_i)}}$$

$$\boldsymbol{p(C = 0|X) = \frac{exp(w_0 + \sum_{i=1}^{n} w_i X_i)}{1 + exp(w_0 + \sum_{i=1}^{n} w_i X_i)}}$$

This form for $\boldsymbol{P(C|X)}$ leads to a simple linear expression for classification. To classify any given X, Logistic Regression algorithm assign the value C_j that maximize $\boldsymbol{P(C = C_j|X)}$.

Data Sets Used:

The Colon dataset is a collection of gene expression measurements from 62 Colon biopsy samples reported by Alon. It contains 22 normal and 40 Colon cancer samples.The Colon data having 2000 genes. Graphical presentation of the dataset is shown below.

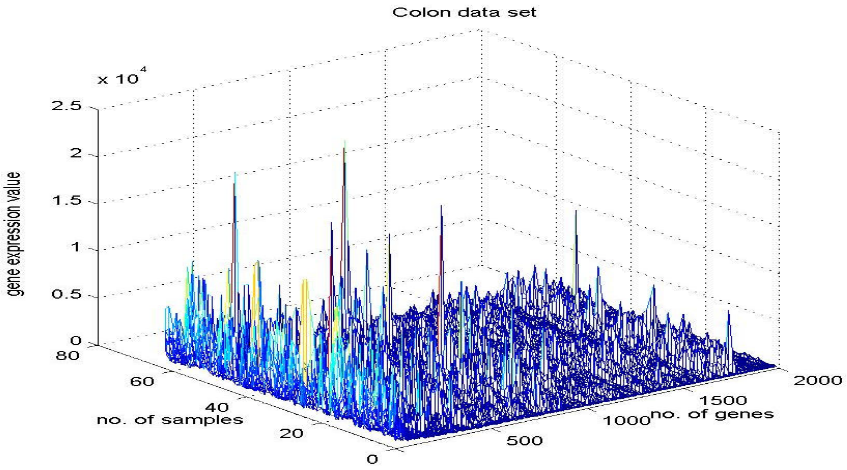

Figure 1 Graphical presentation of Colon dataset

The Leukemia data set is a collection of gene expression measurements from 72 leukemia (composed of 62 bone marrow and 10 peripheral blood) samples with 7129 genes. It contains 47 samples of acute lymphoblastic leukemia (ALL) and 25 samples of acute myeloblastic leukemia (AML). Here we divide the whole dataset into two subsets: training set contains 27 ALL samples, 11 AML samples and test set contains 20 ALL and 14 AML samples.

Results for Colon dataset:

Study of colon dataset is also done using 10-fold cross-validation. Here Bayesian logistic regression algorithm outperforms all other classification algorithms used in the study. However performance of naïve classifier is poor with 53% accuracy rate and also having highest value of absolute relative error. C4.5's performance is also better than the performance of other algorithms except Bayesian logistic regression's performance. Value of absolute relative error is greater than 50% for almost all the algorithms. Only two C4.5 and Bayesian logistic regression have less value of absolute relative error. Table given above shows the correctly classified, incorrectly classified samples with average accuracy and absolute relative error. LMT tree uses D00860 gene for class separation in linear model.

Classification for Colon dataset with 62 samples and 2000 attributes using 10-fold cross validation					
S.No.	Algorithm	Correctly classified	Incorrectly classified	Accuracy Percentage	Absolute Relative Error
1.	CART	47	15	75.8065	65.0738
2.	C4.5	51	11	82.2581	39.7416
3.	LMT	48	14	77.4194	69.9632
4.	Random Forest	48	14	77.4194	75.4522
5.	ADT	47	15	75.8065	57.4395
6.	Naïve Bayesian	33	29	53.2258	101.7067
7.	BayesianLogisticRegression	52	10	83.871	35.0941

Table 1 Classification accuracy table for Colon dataset

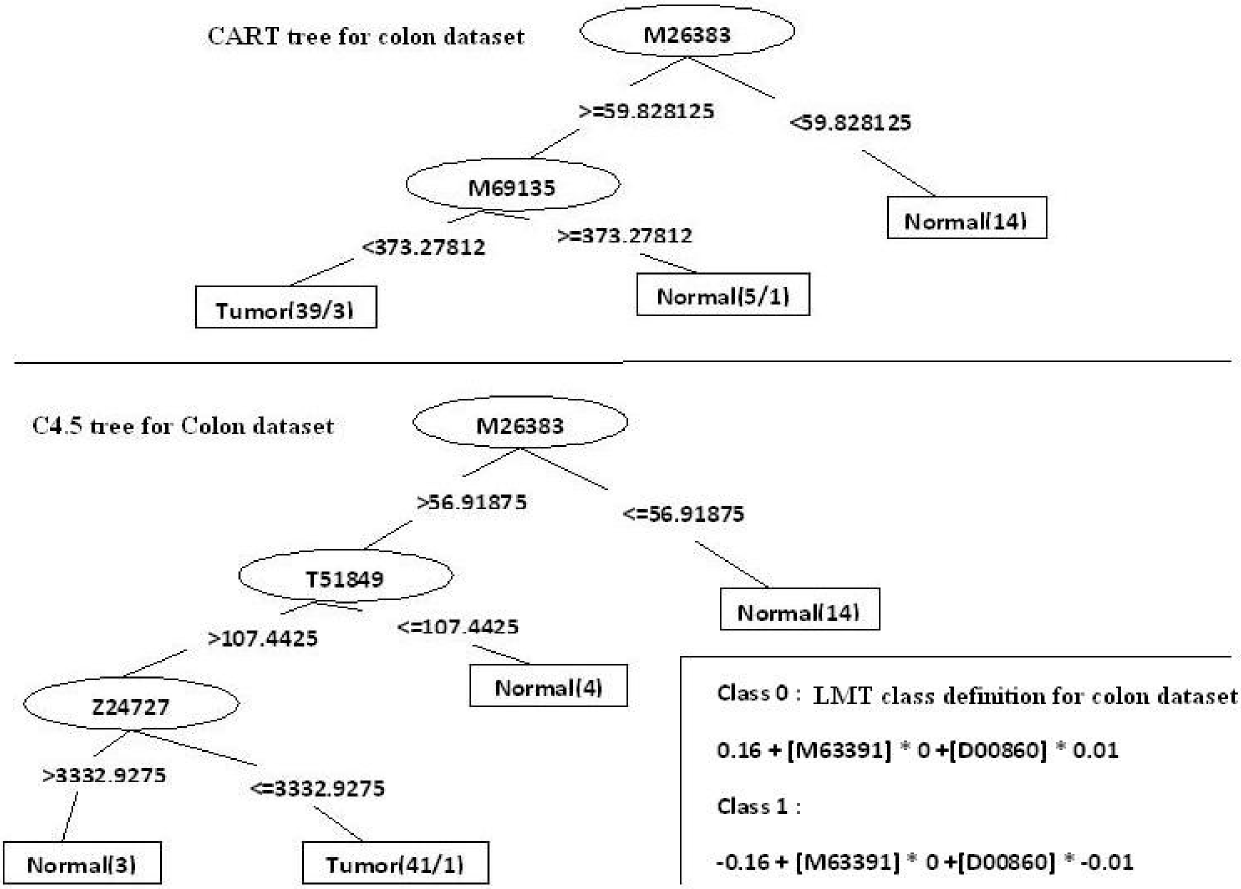

Figure 2 Classification models for Colon dataset

Results for Leukemia dataset:

In case of Leukemia dataset, we used a separate test set for evaluation of classification model with 34 samples instead of 10-fold cross-validation. Here also Bayesian logistic regression algorithm outperforms all other algorithm with accuracy rate 97% and with minimum absolute relative error 6.33%. However performance of naïve Bayesian classifier is reasonably improved with accuracy 88%. Accuracy rate of CART, C4.5 and ADT is same but ADT have low value of absolute relative error. Accuracy of Random Forest algorithm is lowest with higher value of absolute relative error in this case.

Tree generated by Random Forest tree, C4.5 and ADT are shown below. Gene X95735 is most imprtant for the classification of this dataset as shown by C4.5 and ADT algorithm. RandomForest Tree generates a larger tree in comparision to C4.5 and ADTree.

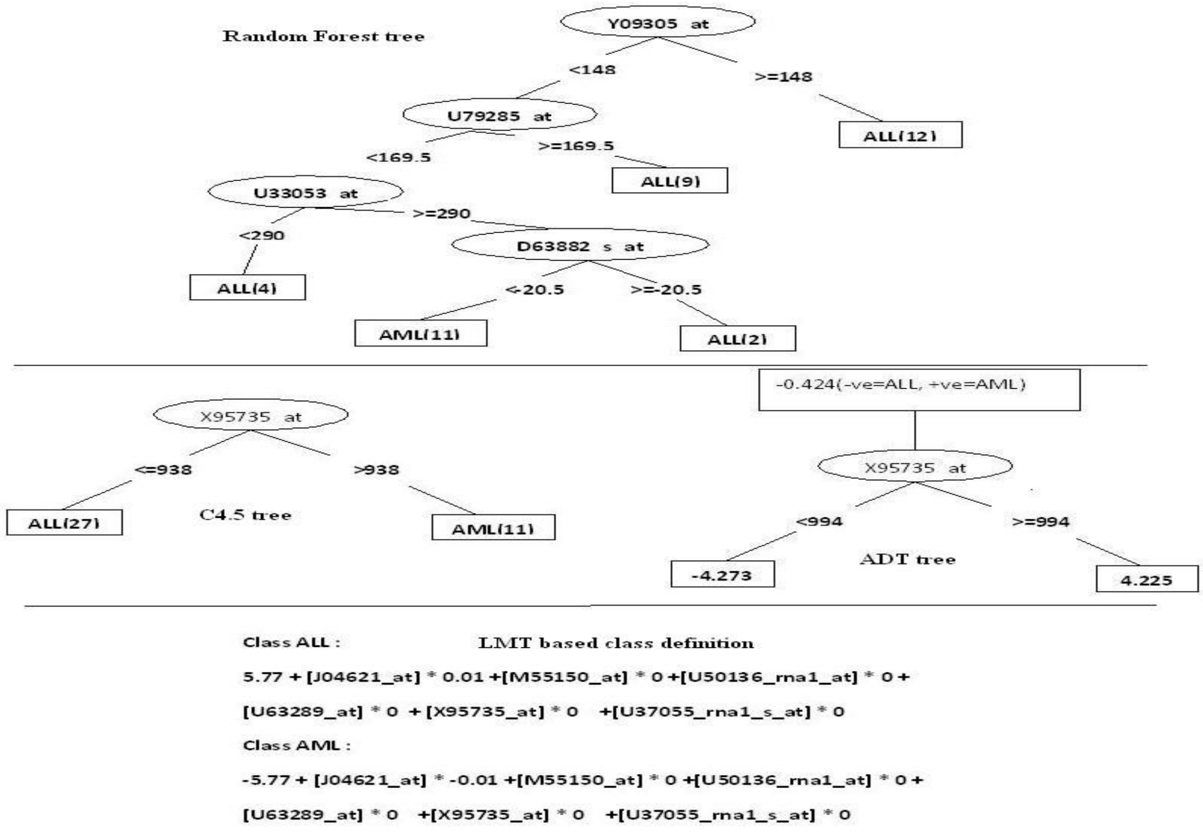
Random Forest tree
Y09305 at
<148
>=148
ALL(12)
U79285 at
<169.5
>=169.5
ALL(9)
U33053 at
<290
>=290
ALL(4)
D63882 s at
<-20.5
>=-20.5
AML(11)
ALL(2)
X95735 at
<=938
>938
ALL(27)
C4.5 tree
AML(11)
-0.424(-ve=ALL, +ve=AML)
X95735 at
<994
>=994
-4.273
ADT tree
4.225
Class ALL :
LMT based class definition
5.77 + [J04621_at] * 0.01 +[M55150_at] * 0 +[U50136_rna1_at] * 0 +
[U63289_at] * 0 + [X95735_at] * 0 +[U37055_rna1_s_at] * 0
Class AML :
-5.77 + [J04621_at] * -0.01 +[M55150_at] * 0 +[U50136_rna1_at] * 0 +
[U63289_at] * 0 +[X95735_at] * 0 +[U37055_rna1_s_at] * 0

Figure 3 Classification models for Leukemia dataset

Classification for Leukemia dataset with 7130 attributes using a training set containing 38 samples and a test set containing 34 samples					
S.No.	Algorithm	Correctly classified	Incorrectly classified	Accuracy Percentage	Absolute Relative Error
1.	CART	31	3	91.1765 %	18.9873 %
2.	C4.5	31	3	91.1765 %	18.9873 %
3.	LMT	29	5	85.2941 %	32.8457 %
4.	Random Forest	24	10	70.5882 %	69.6203 %
5.	ADT	31	3	91.1765 %	21.4818 %
6.	Naïve Bayesian	30	4	88.2353 %	25.3165 %
7.	BayesianLogistic Regression	33	1	97.0588 %	6.3291 %

Table 2 Classification accuracy table for Leukemia dataset

Summary:

Comparison of the classification techniques including C4.5, CART, Random Forest, ADT, LMT, Naïve Bayesian and Bayesian Logistic Regression over different cancer dataset shows that Bayesian Logistic Regression method outperforms the remaining methods. Accuracy of this algorithm is better than the accuracy of other algorithms. However accuracies of C4.5 and ADT algorithms are comparable. Relative absolute error of Random Forest is high for colon and leukemia dataset. However in case of colon dataset, accuracy rate is not high for these algorithms and Naïve Bayesian algorithm has least accuracy rate.

References:

[1] Alexander Genkin, David D. Lewis, David Madigan (2004). Large-scale bayesian logistic regression for text categorization, 2004.

[2] Alizadeh A., Eisen M.B, Davis R.E, et al. *Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. Nature.* 403(6769):503–511,2000.

[3] Bentley J. L. and Friedman J.H., Fast algorithms for constructing minimal spanning trees in coordinate spaces. IEEE Transactions on Computers, C- 27(2): 97 – 105, February 1978.

[4] Cheeseman P., Stutz J.:Bayesian Classification(AutoClass): theory and Results. Advances in Knowledge Discovery and Datamining, 153-180,1996.

[5] Genkin A., Lewis D. Large-Scale Bayesian Logistic Regression for Text Categorization. http://www.stat.rutgers.edu/˜madigan/BBR/, 2004.

[6] Duda, P. E. Hart and D. G. Stork, Pattern Classification, Wiley, New York, 2001.

[7] Fort G., Lambert S., *"Classification using partial least squares with penalized logistic regression"*, England: Bioinformatics-Oxford, 2005.

[8] Freund, Y., Mason, L.: The alternating decision tree learning algorithm. In: Proceeding of the Sixteenth International Conference on Machine Learning, Bled, Slovenia, 124-133, 1999.

[9] Golub T.R, Slonim D.K, Tamayo P, et al. *Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, Science.* 286(5439): 531–537, 1999.

[10] Quinlan J.R., *C4.5: Programs for Machine Learning*. Morgan Kaufman, 1993

[11] Breiman Leo, Jerome H. Friedman, Richard A. Olshen, Charles J. Stone. Classification and Regression Trees. Wadsworth International Group, Belmont, California, 1984.

[12] Breiman Leo. Random Forests. Machine Learning. 45(1):5-32, 2001.

[13] Marc Sumner, Eibe Frank, Mark Hall: Speeding up Logistic Model Tree Induction. In: 9th European Conference on Principles and Practice of Knowledge Discovery in Databases, 675-683, 2005.

[14] Niels Landwehr, Mark Hall, Eibe Frank. Logistic Model Trees. Machine Learning. 95(1-2):161-205, 2005.

[15] Nielsen T.O, West R.B, Linn S.C, et al. *Molecular characterisation of soft tissue tumours: a gene expression study. Lancet* 2002.

[16] Forgionne, G, Gangopadhyay, A, Adya, M., 2000. "Cancer Surveillance Using Data Warehouse, Data Mining, and Decision Support Systems". In *Top Health Inf Manage*: pp.21-34, 2000.

[17] Langley P., Iba W., and Thompson K., "An analysis of bayesian classifiers," in *National Conf. on Artigicial Intelligence*, pp. 223–228, 1992.

[18] Langley P., George H. John. *Estimating Continuous Distributions in Bayesian Classifiers*. Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence. pp. 338-345, 1995.

[19] Andrews R., Diederich J., and Tickle A., "A survey and critique of techniques for extracting rules from trained artificial neural networks," *Knowledge Based Systems*, vol. 8, no. 6, pp. 373–389, 1995.

[20] Dietterich T. G., "Ensemble methods in machine learning," *Lecture Notes in Computer Science*, vol. 1857, pp. 1–15, 2000.

[21] Vapnik V., *The Nature of Statistical Learning Theory*. Springer Verlag, 1995.

[22] Cohen W. W., "Fast effective rule induction," in *Proc. of the 12th Intl. Conf. on Machine Learning*, pp. 115–123, 1995.