Health Care System for Lung Disease Classification using Modified Gini

Tin Yu Wai, Win Pa Pa
University of Computer Studies, Yangon
blackandwhite009@gmail.com

Abstract

Data mining is the process of analyzing data from different perspectives and summarizing it into useful information. Classification has been used for predicting medical diagnosis. Human experts in medical field are frequently in great demand. Health care system giving health information may help patient's symptom is serious or not. Computer based diagnostic systems play an increasingly important role in health care. This paper presents the classification of lung disease classification. Modified Gini algorithm is used to classify the lung diseases in the implementation of health care system. Health care system is implemented as a user friendly diagnosis system for the disease related to lung symptoms using expert system approach.

1. Introduction

Supervised methods are methods that attempt to discover relationship between the input attributes and the target attribute. There are many alternatives to represent classifiers. Originally it has been studied in the fields of decision theory and statistics. However, it was found to be effective in other disciplines such as data mining, machine learning, and pattern recognition. Decision trees are also implemented in many real-world applications.

In medicine, it is very difficult to get correct diagnosis because there are many possible diseases in each case. Computer based methods are increasingly used to improve the quality of medical services. A decision tree is a decision support tool that uses a graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. A decision tree is used to identify the strategy most likely to reach a goal. Another use of trees is as a descriptive means for calculating conditional probabilities. Decision tree technique is most widely used among all other classification methods.

This paper presents the Lung Disease classification using Modified Gini index. Gini index uses the method which biases multivalued attributes. When the number of classes are large, and the biases are increased, the Gini-based decision tree method is modified to overcome the known problems, by normalizing the Gini indexes by taking into account information about the splitting status of all attributes. Instead of using the Gini index for attribute selection ratios of Gini indexes are used and their splitting values in order to reduce the biases.

The following of the paper is organized as the following. Section 1 is the introduction, section 2 is related work. In addition, section 3 illustrates the Expert system; section 4 represents decision tree algorithms and section 5 Proposed System. Section 6 is system implementation and Section 7 is the conclusion of the system.

2. Related Work

The decision tree is probably the most widely used approach in many real world application. A number of such split criteria have been proposed in the literature: gini gain, information gain, gain ratio, etc.

Regression is a very important data mining problem. One very important class of regression models is regression trees. Even though they were introduced early in the development of classification trees (CART, Breiman et al. [4]), regression trees received far less attention from the research community. Quinlan [2] generalized the regression trees in CART by using a linear model in the leaves to improve the accuracy of the prediction. The impurity measure used to choose the split variable and the split point was the standard deviation of the predictor for the training examples at the node. Karalilc [1] argued that the mean square error of the linear model in a node is a more appropriate impurity measure for the linear regression trees since data well predicted by a linear model can have large variance. This is a crucial observation since evaluating the variance is much easier than estimating the error of a linear model (which requires solving a linear system). Even more, if discrete attributes are present among the predictor attributes and binary trees are built (as is the case in CART), the problem of finding the best split attribute becomes intractable for linear regression trees since the theorem that justifies a linear algorithm for finding the best split (Theorem 9.4 in [4]) does not seem to apply. To address computational concerns of normal linear regression models, Alexander and Scott [7] proposed the use of simple linear regressors (i.e., the linear model depends on only one predictor attribute), which can be trained more efficiently but are not as accurate.

Torgo proposed the use of even more sophisticated functional models in the leaves (i.e., kernel regressors) [6, 5]. For such regression trees both construction and deployment of the model is expensive but they potentially are superior to the linear regression trees in terms of accuracy. More recently, Li et al. [3] proposed a linear regression tree algorithm that can produce oblique splits1 using Principal Hessian Analysis but the algorithm cannot accommodate discrete attributes.

This system presents modification of Gini-based decision tree method. Instead of using the Gini index for attribute selection as usual, ratios of Gini indexes are used in order to reduce the biases.

3. Expert System

An expert system is a set of programs that manipulates encoded knowledge and reasoning techniques to solve problems in specialized domain that normally requires the knowledge and the abilities of human experts. Expert system includes

- Knowledge base of domain facts and associated rules that the inference engine uses to draw conclusions or expertise about the user's query
- Inference engine that contains procedure or control structure for selecting the appropriate rules from its knowledge base depending on the facts provided by the user
- A natural user interface to respond to the user's request for an advice

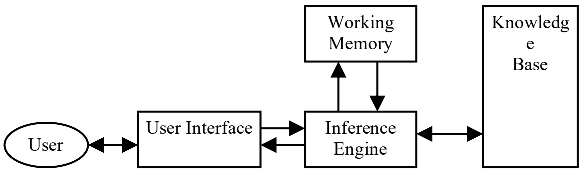

Figure 1. A Typical Expert System

3.1. Data Mining

Data mining refers to extracting or mining knowledge from large amounts of data. It can be also defined as Knowledge Discovery in Databases, or KDD. Data mining is the process of extracting interesting information or patterns from large information repositories such as: relational database, data warehouses, XML repository, etc. Also data mining is known as one of the core processes of Knowledge Discovery in Database (KDD). The main process of KDD is the data mining process. In this process different algorithms are applied to produce hidden knowledge. Then, comes another process called post-processing, this evaluates the mining result according to users' requirements and domain knowledge.

4. Classification

Classification is the task to identify the class labels for instances based on a set of features (attributes). Learning accurate classifiers from pre-classified data is a very active research topic in machine learning and data mining. It classifies data (constructs a model) based on the training set and the values (class labels) in a classifying attribute and uses it in classifying new data. It includes two step process – Model Construction and Model Usage.

Model Construction –

- describing a set of predetermined classes
- Each tuple/sample is assumed to belong to a predefined class, as determined by the class label attribute
- The set of tuples used for model construction is training set
- The model is represented as classification rules, decision trees, or mathematical formulae

Model Usage –

- for classifying future or unknown objects
- Estimate accuracy of the model
- The known label of test sample is compared with the classified result from the model
- Accuracy rate is the percentage of test set samples that are correctly classified by the model
- Test set is independent of training set
- If the accuracy is acceptable, use the model to classify data tuples whose class labels are not known

Many classification and prediction methods have been proposed by researchers in machine learning, expert systems, and statistics.

4.1. Decision Tree

Decision trees are classifiers that represent their classification knowledge in tree form (usually in binary tree form). Each interior node of a decision tree is a test on an attribute. Satisfying that test causes the instance being classified to take one branch out of that node, failing the test causes the instance to take the other branch. A decision tree is used to classify an instance by starting at the root node of the decision tree and following the path the attribute tests dictate until a leaf node is encountered. Each leaf node in a decision tree is a decision, i.e., represents a classification. An instance that ends up at some particular leaf node is classified with the class assigned to that leaf node. A second kind of tree is a class probability tree. This has a vector of class probabilities at each leaf instead of a decision [8].

4.2. Decision Tree by Gini Tree

Gini index builds decision trees from a set of training data using the concept of information entropy. It is a binary tree classifier, which means attributes are partitioned into binary splits. Gini index uses the fact that each attribute of the data can be used to make a decision that splits the data into smaller subsets. The attribute with the highest normalized information gain is the one used to make the decision. The algorithm then recurs on the smaller sub lists.

- If a data set D contains examples from n classes, gini index, gini(D) is defined as

$$gini(D) = 1 - \sum_{j=1}^{n} p_j^2 \qquad (1)$$

where D is the data set,
p_j is the relative frequency of class j in D

- If a data set D is split on A into two subsets D_1 and D_2, the gini index gini(D) is defined as

$$gini_A(D) = \frac{|D_1|}{|D|} gini(D_1) + \frac{|D_2|}{|D|} gini(D_2) \qquad (2)$$

- Reduction in Impurity:

$$\Delta gini(A) = gini(D) - gini_A(D) \qquad (3)$$

- The attribute provides the smallest ginisplit(D) (or the largest reduction in impurity) is chosen to split the node (need to enumerate all the possible splitting points for each attribute)

4.2.1 Drawbacks of Gini Index

Decision tree by Gini Index has following drawbacks.

- Gini index algorithm biases multivalued attributes.
- In addition to having difficulty when the number of classes is large, the method also tends to favour tests that result in equal-sized partitions and purity in all partitions.
- That is why, decision tree by Gini Index may predict the data incorrectly.
- Instead of using the Gini index for attribute selection as usual, modified Gini uses ratios of Gini indexes in order to reduce the biases.

4.3. Decision Tree by Modified Gini Tree

Instead of using the Gini index for attribute selection as usual, ratios of Gini indexes are used in order to reduce the biases. The splitting equation

$$Split_A(D) = 1 - \sum_{j=1}^{m} \left(\frac{|D_j|}{|d|}\right)^2 \qquad (4)$$

Where $Split_A$ (D) = Gini ratio of D with respect to the attribute A
$D_j = D_1, D_2, .. D_m$, the subset values of data values of data set D for the attribute A.
$|D_j|$ = number of records in subset D_j
d = number of records in the data set.

$$giniRatio(A) = \Delta gini(A)/Split_A\ (D) \qquad (5)$$

The attribute with the maximum gini ratio is selected as the splitting attribute

4.3.1.Modified GiniIndex Decision Tree Algorithm

```
Input: The training database D
Output: A decision tree
1. Create a node N
    i.  If D are all of the same class C then return
        N as a leaf node with the class C.
    ii. If D has no Non-label attribute then return N
        as a leaf node with the most common class.
2. Select an attribute, say A, with the highest Gini
   ratio value. Label node N with A.
3. Partition the database D into subsets $D_1,D_2, ..D_m$
   with respect to the attribute A.
4. For each value $a_i$ of A Grow a branch from node
   N for the condition $a_i$
    i.  If $D_i$ is empty then attach a leaf labeled with
        the most common class in D.
    ii. Else attach the node returned by Modified
        Gini Index($D_i$)
  where ,
    $a_i$= $i^{th}$ attribute value of an attribute A
```

5. System Architecture

This system presents Classification of Lung diseases using Modified Gini Index decision tree algorithm. Patient records (with lung diseases) are stored in the training database to build the decision tree. Rules are built from the training database using Modified Gini Index algorithm to classify the input samples. The output of decision tree, rules are used in determining the disease of based on patient's symptoms. Patient's symptoms and measurements are entered into the system through user interface. They are applied into rules of decision tree, it will generate resulted disease and reply back to the user.

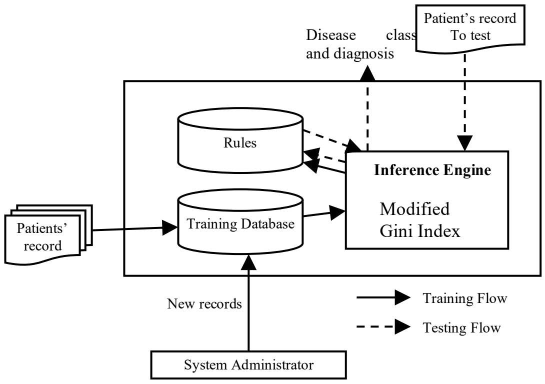

Figure 2. Process Flow of the System

6. System Implementation

This system is implemented using Java programming language. It includes Training process, Testing process and Diagnosis process. In this system, following attributes are used.

6.1. Attributes and Values

Attributes and their values used in the system are shown in Table 1.

Table 1. Attributes and their values

No.	Attribute	Values
1	Age	youth, middle, old
2	Cough	No, Night, Chronic, Severe
3.	Weight Loss	Yes, No
4.	Fatigue	Yes, No
5.	Wheezing	Yes, No
6.	DrugForTB	Yes, No
7.	Cold	Yes, No
8.	Sweat	Yes, No
9.	Chest Pain	No, Mild, Severe
10.	Smoke	No, Sometimes, Severe
11.	Gender	Male, Female
12.	Dysponea	Yes, No
13.	Breathing	Normal, Difficult, Shortness, Rapid
14.	Sneezing	Yes, No
15.	Muscle Pain	Yes, No
16.	Appetite	Yes, No
17.	Head-Ache	No, Mild, Severe
18.	Fever	No, Mild, Severe

Classes: Asthma, Pneumonia, PleuralEffusion, TB, Bronchiectasis

6.2. Process Flow of the System

In this system, decision tree is built using Modified Gini Index algorithm. The process flow of the system is as follows:

- First datasets are read from the training dataset.
- Then Gini (D) is computed based on the probability of class lables of the training samples.
- Then for each attribute in the training samples, $\text{Gini}_A(D)$, ΔGini_A, Split_A (D) and GiniRatio are computed.
- In the computation of $\text{Gini}_A(D)$, since Gini based decision tree is binary tree, therefore, attributes with more than 2 values are grouped to form a two-group attributes in the combination of attributes.
- Then attribute with maximum GiniRatio is selected to form the root of the tree.
- Training data is then split into two groups based on the root attribute and value combination.
- Then splitted datasets is applied into the building tree process again to create the sub tree.
- Those processes are repeated until there is zero Gini (D) or no more attribute left to build the decision tree left or conclusion is reached.
- Finally output decision tree is used as rules to classify the input samples.

6.3. Computations by Modified Gini

There are 18 attributes in the small samples for case study. For each attribute we will compute Gini $_A$ (D), Δ Gini, Split $_A$ (D) and GiniRatio according to equations (1), (2), (3), (4) and (5).

There are two types of disease in the sample training data set. (Asthma and Pneumonia). There are 5 asthma records and 3 pneumonia records. Example dataset is shown in following block.

youth, Chronic, Severe, No, Yes, Yes, No, No, Yes, Severe, No, Female, No, Rapid, Yes, No, No, Mild, Pneumonia
old, Chronic, Severe, Yes, No, Yes, No, Yes, No, Mild, Chronic, Male, Yes, Shortness, Yes, Yes, Yes, Severe, Asthma
middle, Night, No, No, No, Yes, No, Yes, Yes, Mild, Yes, Male, No, Difficult, Yes, No, Yes, No, Asthma
youth, Night, No, No, No, Yes, No, Yes, Yes, Mild, No, Female, Yes, Shortness, No, No, No, Mild, Asthma
middle, Night, Severe, No, Yes, Yes, No, Yes, Yes, Severe, Yes, Male, No, Shortness, Yes, No, Yes, Severe, Asthma
old, No, Mild, No, Yes, Yes, No, No, No, Severe, No, Male, Yes, Shortness, No, No, Yes, No, Asthma
middle, No, Severe, No, Yes, Yes, No, Yes, No, Severe, Chronic, Male, No, Normal, No, No, No, No, Pneumonia
youth, Severe, Severe, No, No, No, No, Yes, Yes, Mild, No, Male, No, Shortness, Yes, Yes, No, Severe, Pneumonia

Modified Gini algorithm is applied above to generate the decision tree as follows.

$$\text{Gini(D)}= 1-\sum_{i=1}^{m} p_i^2$$
$$=1-(5/8)^2-(3/8)^2$$

= 1 – 0.391 – 0.141
= 0.468

Gini (age) D for {youth}, {old, middle}
$=3/8*(1-(2/3)^2-(1/3)^2)+5/8*(1-(4/5)^2-(1/5)^2)$
=0.367

Gini (age) D for {youth}, {old, middle} = 0.367
Gini(age)Dfor{old},{youth,middle}= 0. 0.375
Gini (age) D for {middle}, {old, youth} = 0.467
Minimum Gini (age) {youth}, {old, middle} = 0.367

Δ Gini = 0.4687– 0.367= 0.102

Computing Gini Ratio

Split (age) D = $1 – (3/8)^2 – (5/8)^2$
= 0.469

Gini Ratio = 0.102/0.469= 0.215
Gini Ratio (age) = 0.215
Gini Ratio (cough) = 0.36
Gini Ratio (fever) = 0.36
Gini Ratio (weightloss) = 0.1838
Gini Ratio (fatigue) = 0.0625
Gini Ratio (wheezing) = 0.5102
Gini Ratio (tb) = 0.0000
Gini Ratio (cold) = 0.0278
Gini Ratio (sweat) = 0.00443
Gini Ratio (chestpain) = 0.0625
Gini Ratio (smoke) = 0.25
Gini Ratio (gender) = 0.0278
Gini Ratio (dysponea) = 0.36
Gini Ratio (breathing) = 0.69333
Gini Ratio (sneezing) = 0.00426
Gini Ratio (musclepain) = 0.0278
Gini Ratio (appetite) = 0.5625
Gini Ratio (head-ache) = 0.0278

Max Gini Ratio = 0.69333 for breathing

Example Tree generated by the system is shown in Figure 2.

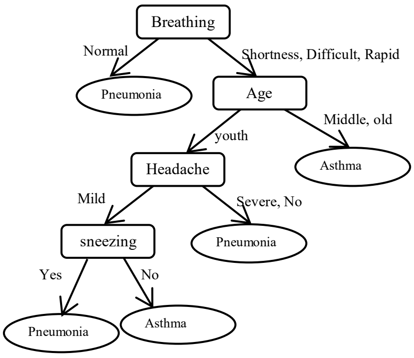

Figure 2. Example Tree

6.4. Experimental Results

The accuracy of a classifier on a given test set is the percentage of test set tuples that are correctly classified by the classifier. The confusion matrix is used to used to compute the accuracy of classifier which can recognize tuples of different classes. Given m classes, a confusion matrix is a table of size m x m. An entry, CMi, j in the first m rows and m columns indicates the number of tuples of class i that were labeled by the classifier as class j. Given two classes, true positives refer to the positive tuples that were correctly labeled by the classifier, while true negatives are the negative tuples that were correctly labeled by the classifier. False positives are the negative tuples that were incorrectly labeled. Similarly, false negatives are the positive tuples that were incorrectly labeled.

Sensitivity and Specificity measures can be used to measure the accuracy. Sensitivity is also referred to as true positive (recognition) rate, propotion of positive tuples that are correctly classified. Specificity is the true negative rate, the proportion of negative tuples that are correctly identified. Precision is also used to access the percentage of correctly classified tuples.

Sensitivity = t_pos / pos. (6)

Specificity = t_neg / neg (7)

Precision = t_pos / (t_pos + f_pos) (8)

Accuracy=Sensitivity(pos/(pos+neg))+
Specificity (neg / (pos + neg)) (9)

where,
t_pos=the number of true positives
pos = the number of positive samples
t_neg = the number of true negative
neg = the number of negative samples
f_pos=number of false positive

This system is tested with 3520 training records and 400 testing records. There are five class lables in this system and detail accuracy of this system for each class label is shown in following table.

Table 1. Experimental results for different class labels

	Class	Records	Accuracy
1	Asthma	987	99.40%
2	Bronchiectasis	258	99.50%
3	PlueralEffusion	253	99.60%
4	Pneunomia	934	99.80%
5	TB	655	99.40%

There are different number of training records for the class labels. All class labels got higher accuracy values as in Figure 3.

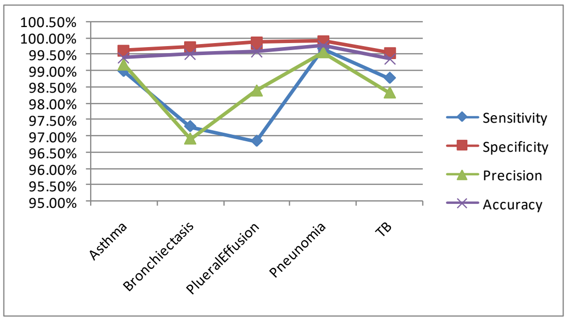

Figure 3. Overal l accuracy of the system

Figure 4 presents the sensitivity values of each class label using modified Gini index. In Figure 5, specificity values are described. The precision and accuracy values are expressed in Figure 6 and Figure 7.

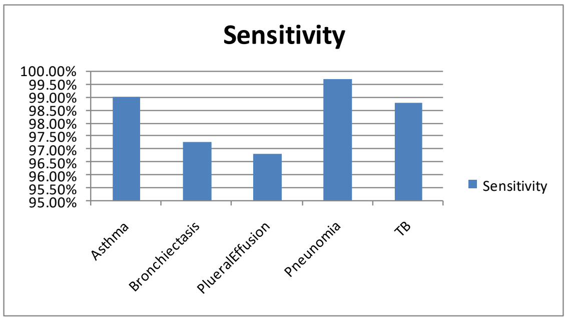

Figure 4. Sensitivity

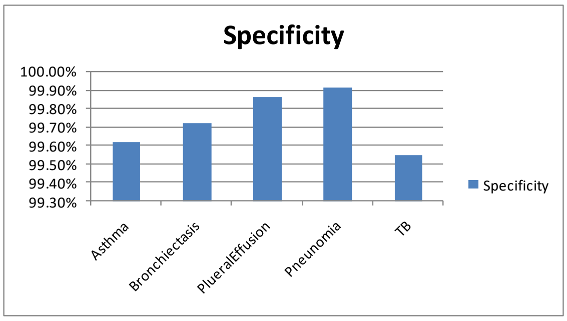

Figure 5. Specificity

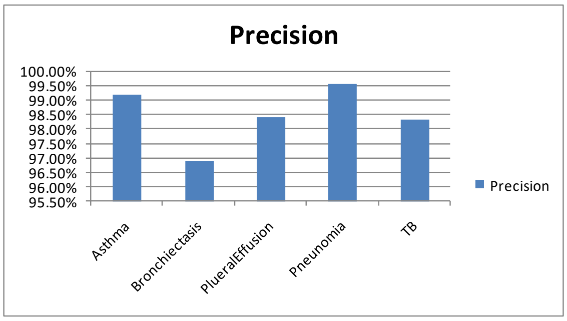

Figure 6. Precision

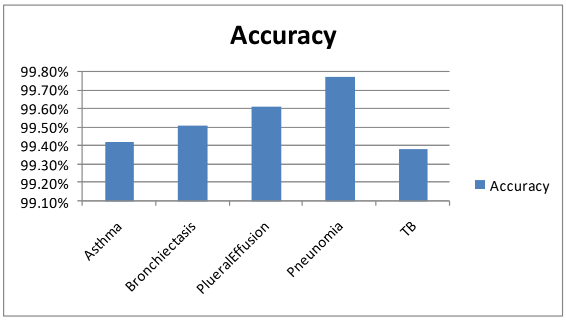

Figure 7. Accuracy

7. Conclusion

In this system, rules can be generated using rule induction method with training data (or) experience without expert's entire help. It saves time because the expert puts data himself and rules can be autogenerated without help of knowledge engineer. These entire rules have only 'IF-THEN' forms so that everybody who uses system can be easily understand. System engineers don't need to understand every step of expert's procedure. They can build knowledge have using experience and data.

8. References

[1] A. Karalic. Linear regression in regression tree leaves. In International School for Synthesis of Expert Knowledge, Bled,Slovenia, 1992.

[2] J. R. Quinlan. Learning with Continuous Classes. In 5th Australian Joint Conference on Artificial Intelligence, pages 343–348, 1992.

[3] K.-C. Li, H.-H. Lue, and C.-H. Chen. Interactive tree-structured regression via principal hessian directions. journal of the American Statistical Association, (95):547–560, 2000.

[4] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. Classification and Regression Trees. Wadsworth, Belmont, 1984.

[5] L. Torgo. Functional models for regression tree leaves. In Proc. 14th International Conference on Machine Learning, pages 385–393. Morgan Kaufmann, 1997.

[6] L. Torgo. Kernel regression trees. In European Conference on Machine Learning, 1997. Poster paper.

[7] W. P. Alexander and S. D. Grimshaw. Treed regression. Journal of Computational and Graphical Statistics, (5):156–175, 1996.