Defining User Knowledge Level by Using Decision Tree Induction Approach

Po Po Zin, Hnin Hnin Aye
Computer University, Myeik
popozin900@gmail.com, hnin2aye@gmail.com

Abstract

This paper describes the method to classify user's knowledge level using decision tree induction. Decision trees can easily be converted to classification rules by using decision tree induction. This system is to estimate classifier accuracy that is important to evaluate how accurately a given classifier will label future data. In this paper, we present the classification of training data in which the resulting classifier is a decision tree induction. Decision Tree method for classification is exploited to identify user knowledge level after they proceed to learn lectures. As a result, user can know their knowledge level after learning and they can also test their knowledge level.

Keywords: Classification, data mining, decision tree, attribute, entropy.

1. Introduction

Data Mining is the search for relationships and patterns that exist in large database but are hidden among the vast amount of data. Classification is the process of finding the common properties among different entities and classification of object is based on the set of data features used and strongly affects classifier design [1].

The Decision Tree is one of the most popular classification algorithms in current use in Data Mining. Decision tree classifiers are found the widest applicability in the large-scale data mining environments [5].

Computer provides users with the opportunity to create new learning environments. It is a user friendly system that user can learn and test without any assistance query human test. This paper demonstrates users to study lectures and examine in a more efficient method. Then, this paper describes user's knowledge level by decision tree induction method.

The main task performed in this system is using decision tree induction methods to the given values of attributes of an unknown object to determine appropriate classification according to decision tree rules. Test attributes are selected on the basis of a heuristic or statistical measure (e.g. information gain measure). Such a measure is referred to as an attribute selection measure or a measure of the goodness of split. This paper examines the decision tree algorithm and implements using dot net programming language [7].

2. Motivation

Classification was the process of finding the common properties among different entities and classifying them into classes. The classification of object was based on the set of data features used and strongly affects classifier design. In this paper, decision tree algorithm is commonly used for gaining information for the purpose of decision making. This paper presents users' information to describe classification rule and decision tree induction algorithm. In this algorithm, an entropy based attributes selection measure is used to select the test attribute at each node in the tree. This intends for users to make a quick reference for decision making on the training data [4].

3. Classification by Decision Tree Induction

A decision tree is a flow-chart-like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and leaf nodes represent classes or class distributions. The top-most node in a tree is the root node. Internal nodes are denoted by rectangles, and leaf nodes are denoted by ovals.

Unknown samples can be classified by testing attributes against the tree. The path traced from root to leaf holds the class prediction for that sample.

There are two steps in decision tree induction. They are model construction and model usage [5].

3.1. Model Construction

Describe 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 training set. The model is represented as classification rules, decision trees, or mathematical formulae [2].

3.2. 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, otherwise over-fitting will occur.

Decision tree starts with a root node on which it is for users to take actions. Decision tree is a classifier in the form of a tree structure, where each node is either:

A Leaf Node - Indicates the value of the target attribute (class) of examples, or

A Decision Node - Specifies some test to be carried out on a single attribute-value, with one branch and sub-tree for each possible outcome of the test.

3.3. Decision Tree Induction Algorithm

The algorithm for decision tree induction is as follow:

Input : The training samples, samples, represented by discrete-valued attributes; the set of candidate attributes, attributes-list.

Output : A decision tree.

Method :

```
(1)  create a node N;
(2)  if sample are all of the same class, C then
(3)      return N as a leaf node labeled with the class $C_i$
(4)  if attribute-list is empty then
(5)      return N as a leaf node labeled with the most common class in samples; // majority voting
(6)  select test-attribute, the attribute among attribute-list with the highest information gain;
(7)  label node N with test-attribute;
(8)  for each known value $a_i$ of test-attribute // partition the samples
(9)      grow a branch from node N for the condition test-attribute = $a_i$ ;
(10)     let $s_i$ be the set of samples in samples for which test-attribute = $a_i$ // a partition
(11)     if $s_i$ is empty then
(12)         attach a leaf labeled with the most common class in samples;
(13)     else attach the node returned by Generate_decision_tree($s_i$, attribute-lists-test-attribute); [3]
```

The basic algorithm for inducing a decision tree from the learning sample set is as follows: [2]

- Initially the decision tree is a single node representing the entire training set.
- If all samples are in the same class, this node becomes a leaf and is labeled with that class label.
- Otherwise, an entropy-based measure, "information gain", is used as a heuristic for selecting the attribute which best separates the samples into individual classes (the "decision attribute").
- A branch is created for each value of the test attribute and samples are partitioned accordingly.
- The algorithm advances recursively to form the decision tree for the sub-sample set at each partition. Once an attribute has been used, it is not considered in descendent nodes.
- The algorithm stops when all samples for a given node belong to the same class or when there are no remaining attributes.

4. System Design

The flowchart of the system is described in figure 1. Each user has a specific allowance in using the system. If the user is new, he/she needs to be registered. The users can study the desired courses. The users can take the exam. After taking the exam, the system will display the result of each user by using decision tree induction algorithm. In this way, the user can be distinguished basic or intermediate or high knowledge level by using this system.

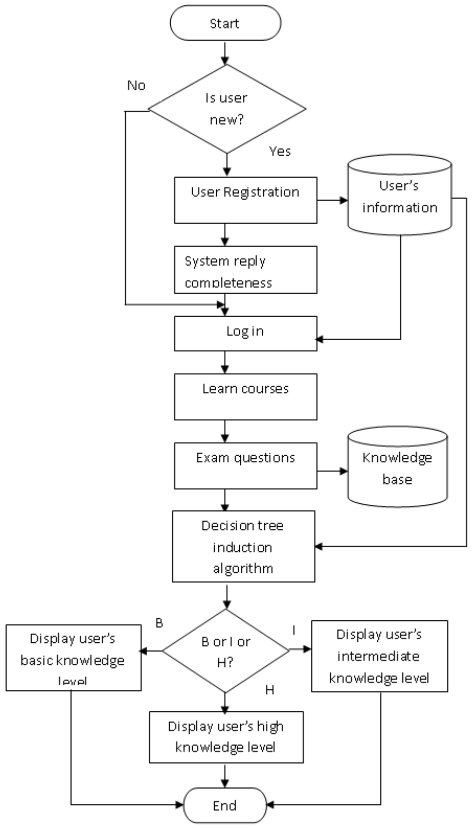

Start
Is user new?
No
Yes
User Registration
User's information
System reply completeness
Log in
Learn courses
Exam questions
Knowledge base
Decision tree induction algorithm
B or I or H?
B
I
H
Display user's basic knowledge level
Display user's intermediate knowledge level
Display user's high knowledge level
End

Figure 1. System Design

The abbreviations in the diagram are specified as follows:

B = Basic
I = Intermediate
H = High

5. Implementation of the System

5.1. Attribute Selection Measure

The information gain measure is used to select the test attribute at each node in the tree. Such a measure is referred to as an attribute selection measure. The attribute with the highest information gain is chosen as the attribute for the current node. In order to define information gain precisely, firstly to discuss the entropy.

$$I\,(s_1, s_2, \ldots, s_m) = -\sum_{i=1}^{m} p_i \log_2 (p_i), \qquad (1)$$

Where,
I =a set consisting of data samples
s_i =number of data samples
p_i =probability that an arbitrary belongs to class C_i

Let attribute A have v distinct values $\{a_1, a_2\ldots, a_v\}$. Attribute A can be used to partition S into v subset, $\{s_1, s_2,\ldots, s_v\}$, where s_j contains those samples in S that value of a_j of A. The entropy, expected information based on the partitioning into subsets of A is given by

$$E(A) = \sum_{j=1}^{m} \frac{s_{1j} + \cdots + s_{mj}}{s}\, I\,(s_1, s_2, \ldots\ldots, s_m) \qquad (2)$$

$$Gain\ (A) = I\,(s_1, s_2, \ldots\ldots, s_m) - E(A) \qquad (3)$$

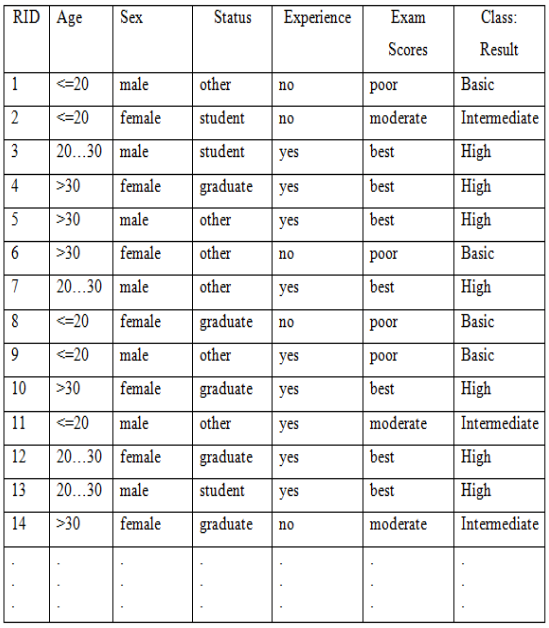
Gain (A) is the expected reduction in entropy caused by knowing the value of attribute (A). The algorithm computes the information gain of each attribute. The attribute with the highest information gain is chosen as the test attribute for the given set [4].

5.2. Experimental Results

Table 1 shows the sample training data set for solving classification problems. First, a training set consisting of records whose class labels are known must be provided. The training set is used to build a classification model, which is subsequently applied to the test set, which consists of records with unknown class labels. Most classification algorithms seek models that attain the highest accuracy, or equivalently, the lowest error rate when applied to the test set.

In this paper, we present the classification of training data in which the resulting classifier is a decision tree induction.

Table 1. Sample Training Data Set



RID	Age	Sex	Status	Experience	Exam Scores	Class: Result
1	<=20	male	other	no	poor	Basic
2	<=20	female	student	no	moderate	Intermediate
3	20…30	male	student	yes	best	High
4	>30	female	graduate	yes	best	High
5	>30	male	other	yes	best	High
6	>30	female	other	no	poor	Basic
7	20…30	male	other	yes	best	High
8	<=20	female	graduate	no	poor	Basic
9	<=20	male	other	yes	poor	Basic
10	>30	female	graduate	yes	best	High
11	<=20	male	other	yes	moderate	Intermediate
12	20…30	female	graduate	yes	best	High
13	20…30	male	student	yes	best	High
14	>30	female	graduate	no	moderate	Intermediate
. . .	. . .	. . .	. . .	. . .	. . .	. . .

Consider the sample training data set in Table 1 as an example. We use total records about 200 data as training data set and about 100 data set is reserved as the test set.

The training set is used to derive the classifier, whose accuracy is estimated with the test set. When classifying the training data set, whose classifier accuracy is about hundred percent, but we classify the test set, whose accuracy is nearly seventy-five percent. Estimating classifier accuracy is important in that allows one to evaluate how accurately a given classifier will label future data, that is, data on which the classifier has not been trained.

The tree has three types of nodes:

- A root node that has no incoming edges and zero or more outgoing edges.
- Internal nodes, each of which has exactly one incoming edge and two or more outgoing edges.
- Leaf or terminal nodes, each of which has exactly one incoming edge and no outgoing edges.

In a decision tree, each leaf node is assigned a class label. The non-terminal nodes, which include the root and other internal nodes, contain attributes test conditions to separate records that have different characteristics. In figure 2 the attribute age is selected as the root node since its information gain is the largest among all of the attributes. The result is not so dependent on the value of a single attribute age, but instead, depends upon the combined values of a set of attributes.

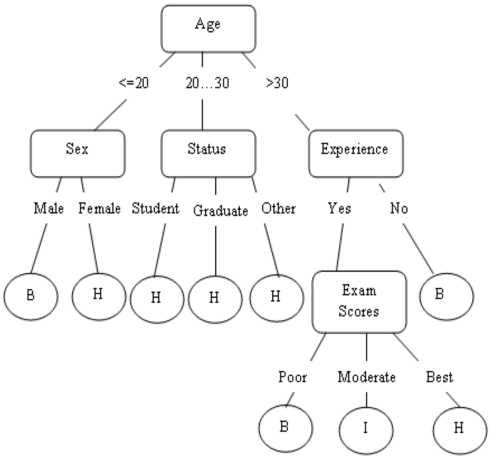


Figure 2. A Complex Decision Tree Describing Training Data

Where,
H= High, B= Basic, I = Intermediate

A compact decision tree is preferable since it is more general and its predictive power is often higher than that of a complex decision tree. Figure 3 a compact decision tree describing of training data.

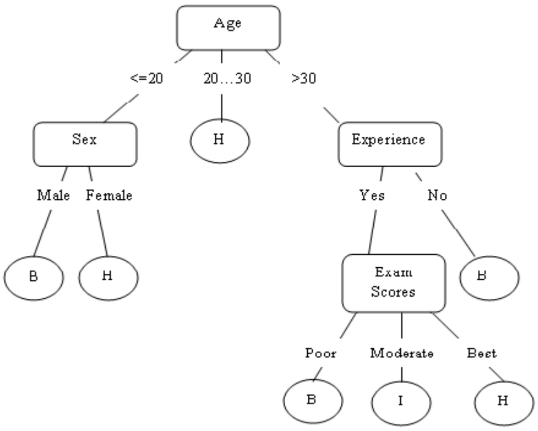


Figure 3. A Compact Decision Tree Describing of Training Data

The following nine steps compute the information gain of attribute age, sex, status, experience and exam scores. The attribute with the highest information gain is chosen as the test attribute for the given set.

Step 1:

Current node C = root node of the tree.

Step 2 and 3:

Entropy of the node C = E(C) = - (4/14) log2 (4/14) - (4/14) log2 (4/14) - (6/14) log2 (6/14) = 1.557

Step 4:

Entropy of the partial tree based on the age attributes:
For age = “<=20’’
E (age) = - (3/5) $\log_2$ (3/5)-(2/5) $\log_2$ (2/5) - (0/5) $\log_2$ (0/5) = 0
For age = “20 … 30’’
E (age) = - (0/4) $\log_2$ (0/4)-(0/4) $\log_2$ (0/4) - (4/4) $\log_2$ (4/4) =0
For age = “10…20 ’’
E (age) = - (1/5) $\log_2$ (1/5)-(2/5) $\log_2$ (2/5) - (2/5) $\log_2$ (2/5) = 1.522
E (age) = (5/14)*E (age) + (4/14)*E (age) + (5/14)*E (age) = 0.544

Step 5:

Information gain due to the partition by the age attributes:
G (age) = E(C) –E (age) = 1.013

Step 6:

Similarly, the information gains due to the partition by the sex, status, experience and exam scores attributes, respectively, are:
G (sex) = 0.573
G (status) = 0.442
G (experience) = 0.429
G (exam scores) = 0.557

Step 7:

The age attribute is selected as the classifying attribute for the current node C since its information gain is the largest among all of the attributes.

Step 8:

The resulted partial decision tree is:

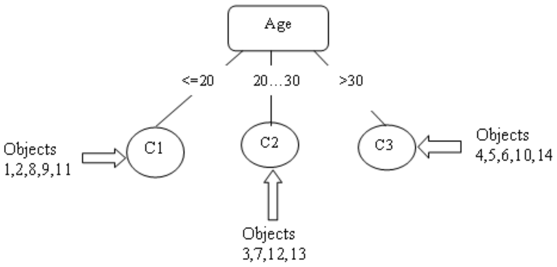


Figure 4. The Resulted Partial Decision Tree

Step 9:

The analysis continues for the node C1 and C3 until all of the leaf nodes are associated with objects of the same class.

The resulted final decision tree is:

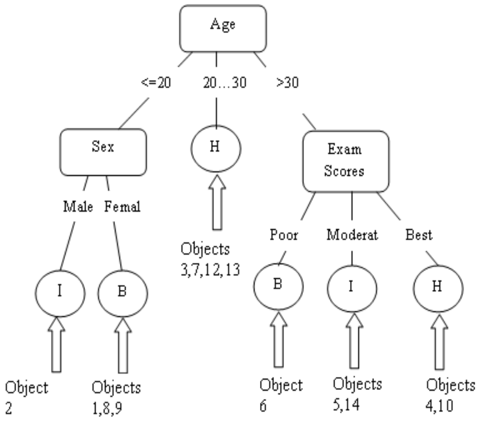


Figure 5. The Resulted Final Decision Tree

Figure 5 refers to the resulted final decision tree by using Table 1. The result is depend on the whole attributes and decides the level of knowledge for the users. In this figure, there are types of sex for the age of under 20. If the sex is male, the user's knowledge level is Intermediate and the objects ID is 2 and for female, the level is Basic and the objects ID are 1, 8 and 9 according to the table 1. There is High level between 20 to 30, and the objects ID are 3, 7, 12 and 13. There are three types of level for the age of over 30. If the exam score is poor, the level is Basic and the objects ID is 6, and for moderate, the level is Intermediate and, objects ID are 5 and 14. And then for high, the level is High and the objects ID are 4 and 10.

6. Conclusion

To construct a supervised classification system, one requires a data set of labeled examples with which to train and test the system. Each example is made up of a class label.

This paper uses decision tree induction in data mining classification algorithms to get useful information to decision-making out of user's knowledge level. It reduces time consuming, and can easily determine users' knowledge level. By this means, this system is useful for everyone who wants to entrance examination about the computer courses of learning.

7. References

[1] D.A Kem,"**Knowledge Discovery and Data Mining",** Newport Beach, USA, 1997.

[2] Jiawei Han and Micheline Kamber, "**Data Mining: Concepts and Techniques",** Department of Computer Science , University of Illinois at Urbana-Champaign.

[3] Khin Kywe Kywe, Aye Aye Thein, "**Evaluation of Classification Algorithms",** University of Computer Studies, Mandalay, 2008.

[4] Manpreet Singh, Parminder Kaur Wadhwa and Parvinder Singh Sandhu, "**Human Potein Function Prediction using Decision Tree Induction",** Deptt. Of CSE & IT, Guru Nanak Dev Engineering College, Ludhiana, Punjab, INDIA.

[5] Micheline Kamber, Lara Winstone, Wan Gong, Shan Cheng, Jiawei Han, "**Generalization and Decision Tree Induction: Efficient Classification in Data Mining",** Database Systems Research Laboratory, School of Computing Science, Simon Fraser University, B.C., Canada V5A 1S6.

[6] Minos Garofalakis, Dongjoon Hyun, Rajeev Rastogi and Kyuseok Shim, "**Efficient Algorithms for Constructing Decision Trees with Constraints".**

[7] Myo Myo Than Naing, Tin Htar Nwe, **"Decision Making System Using Decision Tree Induction Algotithm"**, University of Computer Studies, Yangon.