General Criteria on Building Decision Trees for Data Classification

Yo-Ping Huang
Department of Electrical Engineering
National Taipei University of Technology
Taipei, Taiwan 10608
Tel: +886-2-27712171ext 1002
yphuang@ntut.edu.tw

Vu Thi Thanh Hoa
Department of Electrical Engineering
National Taipei University of Technology
Taipei, Taiwan 10608
Tel: +886-912450745
t7318111@ntut.edu.tw

ABSTRACT
Decision trees have been widely and successfully applied to data mining and machine learning for data classification. One of the critical components in a decision tree algorithm is the criterion used to select which attribute will become a test attribute in a given branch of the tree. Several algorithms, such as ID3, C4.5, and CART, are investigated and compared for determining attributes as the root and the branches in decision trees. This paper presents the issues of traditional ID3 algorithm that directly affect building decision trees. The goal is to select critical factors in ID3 algorithm that result in an efficient decision tree for classification applications. Examples are given to illustrate what factors may affect the construction of decision trees.

Keywords
Classification, ID3 algorithm, decision tree, information gain.

1. INTRODUCTION
Data classification is the process that finds the common characteristics among a set of objects in a database and classifies them into different classes according to a classification model. Up to the present, the development of classification has made great achievements, and various kinds of classification technologies and theories will continue to emerge. It has been applied in many areas, such as fraud detection, target marketing, weather prediction, performance prediction, manufacturing, and medical diagnosis.

Classification approaches mainly include decision tree, neural network, Bayesian classification, case-based reasoning, genetic algorithms and rough sets method, in which decision tree is applied most commonly. A decision tree based classification method, such as [1], [2], is a supervised learning method that constructs decision tree from a set of samples.

Generally, a decision tree is a flow-chart-like tree structure, in which each leafnode is assigned a class label. The top node in the tree, called the root, contains all training samples that are to be divided into classes. All nodes except the leaves are called decision nodes, since they specify decision to be performed at this node based on a single feature. Each decision node has a number of children nodes, equal to the number of values that a given feature assumes. All decision tree algorithms are based on Hunt's fundamental algorithm of concept learning [5], [6]. This algorithm embodies a method used by humans when learning simple concepts, namely, finding key distinguishing features between two categories, represented by positive and negative (training) examples. Hunt's algorithm is based on a divide-and-conquer strategy. The task is to divide the set S, consisting of n samples belonging to c classes, into disjoint subsets that create a partition of the data into subsets containing samples from one class only. Decision tree has many advantages, such as its fast speed in tree construction and high accuracy in classification that makes it attractive in data mining applications.

A typical learning system, ID3 (Iterative Dichotomiser 3) [1], was proposed by Quinland in 1986 when building a decision tree for a given training dataset. For ID3 algorithm, decision tree is constructed in a top-down recursive, and uses an information theoretic approach aimed at minimizing the expected number of tests to classify an object. The basic idea of the induction algorithms is to ask questions whose answers provide the most information. The basic strategy is to choose splitting attributes with the highest information gain first. The amount of information associated with an attribute value is related to the probability of occurrence. ID3 was designed for the other end of the spectrum, where there are many attributes and the training set contains many objects, but where a reasonably good decision tree is required without much computation.

Up to now, ID3 algorithm has been widely used in real world applications [9], [10], [11]. It has played an important role to enhance the performance of induction algorithm. Thus, many researches purely apply ID3 to make decisions, but not consider the impact of some factors on the algorithm. In this study, we will concentrate on the problem. We analyze four elements in detail by using the real data in the examples. The results affirm that in some cases, when one of the factors changes, one needs to rebuild the decision tree structure.

The origin of four elements emerges from the actual situations which affect the decision actions. Now, we will see each by an example to define the four elements in their own logicality.

ICIS 2009, November 24-26, 2009 Seoul, Korea

First, in a clothes shop, one can classify the clothes according to gender or age groups. If relying on gender, all shop's clothes will be classified to two categories (classes or attributes): clothes for male and female. If considering from age groups, the clothes can be classified to three categories (clothes for kids, clothes for youth, and for middle age). According to other purposes of classification, the calculated information gains are different because the entropy is related to information gain. The higher the entropy, or uncertainty, of some data is, the more information is required in order to completely describe that data.

Continuously, we assume that the shop has 50 suits of clothes. Applying the first classification norm, we have 20 suits for male and 30 suits for female whereas if the second classification norm is used, we have 10 suits for kids, 15 suits for youth and 25 suits for middle age. Accordingly, the entropy will be 0.971 bits and 1.485 bits, respectively. The conclusion is that the number of classes (or attributes) of one element has effect on decision actions.

In the second situation, we carry out an investigation of academic performance on electrical engineering students. We will explain the effect when changing the number of samples in sub-attributes. Assume that, at the beginning we take two classes randomly A and B, with class A having 32 students (2 females and 30 males) and class B having 33 students (8 females and 25 males). The entropy will be 0.619 bits. However, when we replace class A with class C (25 students with 5 females and 20 males), the entropy will be changed to 0.768 bits for this case. From this example, we see that the number of samples in sub-attribute is one factor we have to consider in decision action.

Last, the number of samples for each class in one sub-attribute is also a basic factor. Let us consider the underlying example. We have another academic performance survey in one department. We split randomly all 300 students in this department into 2 groups and each group has 150 students. At the first time, we pick 20% of students from each group. We see that there are 6 males and 54 females. Then the ratio between male and female students in this picked group is 1/9. After the first test, we let students return to their groups. In the second test, we repeat the process by randomly picking 20% of students from each group. Calculating the ratio between male and female students again, we have the ratio of 1/5 (10 males and 50 females). The difference between the ratios of two tests leads to the unbalance survey results.

From the above three simple examples, we have shown 4-main elements which have effect on building decision tree. They are the number of sub-attributes in each attribute, the number of classes, the number of samples in each sub-attribute, and the number of samples for each class in one sub-attribute. The four factors will be continuously clarified in the following parts.

This paper is organized as follows: Section 2 reviews ID3 algorithm and considers the applied example. Section 3 discusses issues on using ID3 algorithm to find the main factors when building decision trees. Section 4 summarizes a few general remarks of using ID3.

2. ID3 ALGORITHM

The aim of ID3 algorithm is to construct a decision tree that bases on the given categorical attribute values to correctly classify the unknown objects. ID3 algorithm uses a fixed set of examples to build a decision tree and then uses this tree to classify given data samples.

2.1 Overview of ID3 Algorithm

The basic structure of ID3 is iterative [1]. A subset of the training set called the *window* is chosen at random and a decision tree formed from it. If the tree gives the correct answer for all these objects then it is correct for the entire training set and the process terminates. If not, a selection of the incorrectly classified object is added to the window and the process continues. In this way, correct decision trees have been found after only a few iterations for training sets of up to thirty thousand objects described in term of up to 50 attributes. Empirical evidence suggests that a correct decision tree is usually found more quickly by this iterative method than by forming a tree directly from entire training sets.

ID3 algorithm produces a classification tree by learning from training samples. Each training pattern S consists of a set of m input features and an associated class.

We describe the algorithm below where S contains s_i examples of class C_i for $i = \{1,...,m\}$ and A represents the attribute list [3]:

ID3(S, A)

(1). Create a node T.
(2). **If** S consists of samples with all the same class C **then** return T as a leaf node labeled with class C.
(3). **If** A is empty, **then** return T as a leaf node with the majority class in S.
(4). Select test attribute (TA) among the A's with the highest information gain.
(5). Label node T with TA.
(6). For each known value a_i of TA
   (a) Grow a branch from node T for the condition $TA = a_i$.
   (b) Let s_i be the set of samples in S with $TA = a_i$.
   (c) **If** s_i is empty **then** attach a leaf labeled with the majority class in S.
   (d) **Else**, attach the node returned by **ID3**$(s_i, A - TA)$.

According to ID3 algorithm, each non-leaf node of the tree contains a splitting point, and the main task for building a decision tree is to identify an attribute for the splitting point based on the information gain. Information gain can be computed using entropy which is used to measured the amount of uncertainty or surprise or randomness in a set of data.

The expected information needed to classify a sample in S is given by:

$$I(s_1, s_2, ..., s_m) = -\sum_{i=1}^{m} \frac{s_i}{s} \log_2 \frac{s_i}{s} . \qquad (1)$$

A logarithm function to the base of two is used because the information is encoded in bits. $I(s_1, s_2, ..., s_m)$ is just the average amount of information needed to identify the class label of a sample in S. Note that, at this point, the information we have is based solely on the proportions of (samples) of each class. $I(s_1, s_2, ..., s_m)$ is also known as the entropy of S.

In addition, assume the entropy of attribute A with v different values $\{a_1, a_2, ..., a_v\}$:

$$E(A)=\sum_{j=1}^{v}\frac{s_{1j}+...+s_{mj}}{s}I(s_{1j},...,s_{mj}). \quad (2)$$

Therefore, information gained by branching on attribute A:

$$Gain(A)=I(s_1,s_2,...,s_m)-E(A). \quad (3)$$

Gain (A) is the expected reduction in entropy caused by knowing the value of attribute A. The value of *Gain (A)* is the number of bits saved when encoding the target value of an arbitrary member of S by knowing the value of attribute A.

2.2 An Example

In practice, ID3 is applied to many areas and it is very helpful to make decisions.

In this paper, we use an example [4] to illustrate how to calculate the information gain using entropy. Table 1 shows several data congregations of relating guidelines that influence on the degree of weather comfort.

To learn a decision tree (a classifier) from the above data set, the following four attributes of "clothes", "temperature", "humidity", and "windy" can be selected to split the training data.

Table 1. The original dataset

Clothes	Temperature	Humidity	Wind	Class
more	high	prodigious	no	N
more	high	prodigious	great	N
more	high	prodigious	middling	N
normal	high	prodigious	no	P
normal	high	prodigious	middling	P
enough	moderate	prodigious	no	N
enough	moderate	prodigious	middling	N
enough	high	very good	no	P
enough	high	very good	great	N
more	moderate	prodigious	no	N
more	moderate	prodigious	middling	N
enough	moderate	very good	no	N
enough	moderate	very good	middling	N
more	moderate	very good	middling	P
more	moderate	very good	great	P
normal	moderate	prodigious	great	P
normal	moderate	prodigious	middling	P
normal	high	very good	no	P
enough	moderate	prodigious	great	N
normal	high	very good	middling	P

In this example, two distinct values P and N in attribute *Class* denote "*comfortable*" and "*uncomfortable*", respectively; therefore, there are two distinct classes (m=2). There are nine samples of class P and eleven samples of class N. A (root) node T is created for the samples in S. To find the splitting criterion for these samples, we must compute the information gains of the four test attributes and compare the results with each other. We first use Eq.(1) to compute the expected information needed to classify a sample in S:

$$I(p,n)=I(9,11)=-\frac{9}{20}\log_2\frac{9}{20}-\frac{11}{20}\log_2\frac{11}{20}$$
$$=0.993\text{ bits}$$

Next, we need to compute the expected information requirement for each attribute. For each sub-attribute in each attribute, we find the number of P and N samples to apply Eq.(2). For attribute "clothes", there are three sub-attributes: more, normal, enough. The distributions of P and N samples for each sub-attribute of "clothes" are summarized in Table 2.

Table 2. Distributions of the samples in original "clothes"

Clothes			
For clothes = "*more*"	p_1=2	n_1=5	I(p_{11},n_{21})=0.863
For clothes = "*normal*"	p_2=6	n_2=0	I(p_{12},n_{22})=0
For clothes = "*enough*"	p_3=1	n_3=6	I(p_{13},n_{32})=0.592

Using Eq.(2), the expected information needed to classify a sample in S if the examples are partitioned according to "clothes" is:

$$E(clothes)=\frac{7}{20}I(p_1,n_1)+\frac{6}{20}I(p_2,n_2)+\frac{7}{20}I(p_3,n_3)$$
$$=0.509\text{ bits}$$

Therefore, information gained by branching on attribute "clothes" would be:

$$Gain(clothes)=I(p,n)-E(clothes)=0.993-0.509$$
$$=0.484\text{ bits}$$

Similarly, we can find the distributions of remaining three attributes in Table 3.

Table 3. Distributions of the samples in original "temperature," "windy," and "humidity"

Temperature			
For temperature= "*high*"	p_1=5	n_1=4	I(p_{11},n_{21})=0.991
For temperature= "*moderate*"	p_2=4	n_2=7	I(p_{12},n_{22})=0.946
Windy			
For windy= "*no*"	p_1 =3	n_1=4	I(p_{11},n_{21})=0.985
For windy= "*great*"	p_2 =2	n_2=3	I(p_{12},n_{22})=0.971
For windy= "*middling*"	p_3 =4	n_3=4	I(p_{13},n_{23})=1.000
Humidity			
For humidity= "*prodigious*"	p_1 =4	n_1=8	I(p_{11},n_{21})=0.918
For humidity= "*very good*"	p_2 =5	n_2=3	I(p_{12},n_{22})=0.954

Therefore, the results of the gain in information by branching on all four attributes are displayed in decreasing bit values in Table 4.

From the above results, "clothes" attribute has the highest information gain and becomes the splitting attribute at the root node T of decision tree. Branches are grown for each sub-attribute of “clothes”. The samples are then partitioned accordingly, which is described in Fig. 1. The decision tree is formed eventually as shown in Fig. 2.

Table 4. Information gained by branching on four attributes

Attribute	Gain
clothes	**0.484**
temperature	0.027
windy	0.005
humidity	0.061

3. Decisive Factors in Decision Trees

The accuracy of classification done by ID3 algorithm mostly depends on a training dataset. Every different training dataset resulted in different decision trees and different accuracy while classifying the same testing datasets. The advantage of learning a decision tree is that a program, rather than a knowledge engineer, elicits knowledge from an expert.

Relying on the principle of ID3 to build a decision tree, in this paper we find some general factors that affect the structure of decision tree.

Now, we are going to analyze each factor in detail by offering corresponding samples. Note that in the first three cases, we make the effect on only one attribute (three attributes remained unchanged), and always keep the number of samples be nine and eleven in class P and N, respectively.

3.1 The Number of Sub-attributes in Each Attribute

The sub-attribute is always an important concept in splitting attributes. Sub-attributes describe the separate characteristics of each attribute to help effectively classify samples. In the previous example, the “temperature” attribute has two sub-attributes. Now, we will split the “temperature” attribute into three sub-attributes: *high*, *moderate* and *low* but the total of samples in each class remained unchanged. We assign different values to these attributes and employ the above three equations, i.e., Eq.(1), (2) and (3) to get the result as shown in Table 5.

Table 5. Results with respect to the attributes after changing the number of sub-attributes in each attribute

Temperature			
For temperature = “*high*”	p_1=1	n_1=4	I(p_{11},n_{21})=0.722
For temperature = ”*moderate*”	p_2=7	n_2=0	I(p_{12},n_{22})=0
For temperature = “*low*”	p_3=1	n_3=7	I(p_{13},n_{32})=0.544

Attribute	Gain
clothes	0.484
temperature	**0.595**
windy	0.005
humidity	0.061

Compared with the original result in Table 4, we found that gain of temperature in this case is higher. Based on the results in Table 5, the “temperature” attribute has the highest gain among the given four attributes so that it will become the root of tree to replace the “clothes” attribute. Thus, the structure of decision tree must be rebuilt in the same method.

3.2 The Number of Samples in Each Sub-attribute

The unbalance amount of samples in each sub-attribute is the main reason of changing the gain of attributes according to ID3 algorithm. The higher unbalance amount of samples is, the less information gain it provides. To describe this effect, we will consider another case. We make the modification in the number of samples in each sub-attribute for “humidity” attribute. In the original example, “humidity” attribute has two sub-attribute *prodigious* and *very good* with amounts of 12 samples and 8 samples, respectively. We will change the number of samples in these sub-attributes to 7 samples and 13 samples. Then, the result is obtained by employing ID3 algorithm.

Table 6 gives a result for purpose of choosing the attribute that will become the root of tree. Therefore, in this case, “humidity” attribute is assigned as the root of tree and a new tree will be constructed from this root node. That is also the reason of changing decision tree.

Table 6. Results with respect to the attributes after changing the number of samples in each sub-attribute

Humidity			
For humidity = “*prodigious*”	p_1=7	n_1=0	I(p_{11},n_{21})=0
For humidity = ”*very good*”	p_2=2	n_2=11	I(p_{12},n_{22})=0.619

Attribute	Gain
clothes	0.484
temperature	0.027
windy	0.005
humidity	**0.591**

3.3 The Number of Samples for Each Class in One Sub-attribute

In the above two cases, we used some examples to demonstrate that when the amount of sub-attributes in each attribute or the amount of samples for each sub-attribute change, the structure of decision tree will be rebuilt in some situations.

In addition, to find the impact of amount of samples for each class, another situation is considered. In which we change the number of samples in two classes of each sub-attribute but always make sure that total samples for each sub-attribute in “humidity” attribute are always fixed (still 12 and 8 samples for *prodigious* and *very good* sub-attributes, respectively) same as Table 7. This leads to the new decision tree being established with the new root node is “humidity” attribute.

3.4 The Number of Classes

Substituting p_j for s_j / s, p_j is the relative frequency of class j in S or the probability. Thus, Eq.(1) will become:

$$Entropy = \sum_{j=1}^{m} - p_j \log_2 p_j \ . \tag{4}$$

Entropy of a pure table (consisting of single class) is zero because the probability is one and $\log(1) = 0$. Entropy reaches maximum value when all classes in the table have equal probability. The entropy reaches maximum for number of classes *m*, with equal probability $p_j = 1/m$. In this case, maximum entropy is equal to $-m * p_j * \log p_j$. Notice that the value of entropy is larger than one if *m* is larger than two.

We found that the value of entropy of an attribute is also a factor depending on *m*. Now, we will show another example to make this matter clear.

Table 7. Results with respect to the attributes after changing the number of samples for each class in one sub-attribute

Humidity			
For humidity = *"prodigious"*	p_1=1	n_1=11	I(p_{11},n_{21})=0.414
For humidity = *"very good"*	p_2=8	n_2=0	I(p_{12},n_{22})=0

Attribute	Gain
clothes	0.484
temperature	0.027
windy	0.005
humidity	**0.745**

Assume that we classify 20 samples in the original data into three classes as P, N and M with the distribution of attributes in Table 8. We always make sure that the amount of samples in the same class of all attributes are the same. In this case we fix the number of samples in three classes in order being seven, seven and six.

We must redo all steps same as the original example for four attributes. The result shows that "windy" attribute achieves the highest gain that is different from the original example with "clothes" attribute. Consequently, the decision tree also must be reconstructed to fit the new root "windy".

4. CONCLUSION

This paper presents some analyses relying on ID3 algorithm to find the factors that affect the building of decision tree. Four main factors that are mentioned include the number of sub-attributes in each attribute, the number of samples in each sub-attribute, the number of samples for each class in one sub-attribute and the number of classes. When one of these factors is changed, the structure of decision tree must be rebuilt in some cases. A well-constructed decision tree plays a decisive role in classifying unknown data to the appropriate classes.

This study only mentions four most significant factors that are achieved when analyzing the training data table. In the future, we will explore more factors that might also affect on the construction of decision tree and find out the accuracy of the classification on each factor.

5. ACKNOWLEDGMENT

This work is supported by National Science Council, Taiwan under Grants NSC97-2221-E-027-034-MY3.

6. REFERENCES

[1] Quinlan, J.R. 1986. Introduction of decision trees. Machine Learning, vol. 1, 81-106.

[2] Quinlan, J.R., C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers Inc., San Fancisco, CA, USA, 1993.

[3] Han, J. and Kamber, M., Data Mining: Concepts and Techniques, Second Edition, Morgan Kaufmann Publishers Inc., San Fancisco, CA, USA, 2006.

[4] Li, G.-G., Li, Y., Li, F.-C. and Jin, C.-X. 2007. Decision tree inductive learning algorithm based on removing noise gradually. In Proceeding of Int. Conf. on Machine Learning and Cybernetics (Hong Kong, China, Aug. 2007), vol. 5, 2723-2728.

[5] Song, X.-D. and Cheng, X.-L. 2007. Decision tree algorithm based on sampling. In Proceeding of IFIP Int. Conf. on Network and Parallel Computing Workshops (Dalian, China, Sept. 2007), 689-694.

[6] Li, G.-G., Li, Y. and Ren, Y.-H. 2008. Induction algorithm based on statistics theory with Delphi. In Proceeding of Int. Conf. on Wavelet Analysis and Pattern Recognition (Hong Kong, China, Aug. 2008), vol. 2, 799-804.

[7] Samet, S. and Miri, A. 2008. Privacy preserving ID3 using Gini index over horizontally partitioned data. In Proceeding of IEEE/ACS Int. Conf. on Computer Systems and Applications (Doha, Qatar, March/April 2008), 645-651.

[8] Li, Z., He, B. and Chen, X. 2008. Analysis and disposal of pathogenic factors of exogenous affections based on the ID3 decision tree method. In Proceeding of 7th World Congress on Intelligent Control and Automation (Chongqing, China, June 2008), 7509-7514.

[9] Jearanaitanakij, K. 2005. Classifying continuous data set by ID3 algorithm. In Proceeding of Fifth Int. Conf. on Information, Communications and Signal Processing (Bangkok, Thailand, Dec. 2005), 1048-1051.

[10] Yan, K.-W., Zhu, J.-F. and Qiang, S. 2007. The application of ID3 algorithm in aviation marketing. In Proceeding of Int. Conf. on Grey Systems and Intelligent Services (Nanjing, China, Nov. 2007), 1248-1288.

[11] Zhao, F. and Su, H.-S. 2008. A decision tree approach for power transformer insulation fault diagnosis. In Proceeding of 7th World Congress on Intelligent Control and Automation (Chongqing, China, June 2008), 6882-6886.

[12] Zhai, J.-H., Zhang, S.-F. and Li, Z.-Q. 2007. Aggregate operator defined on partition of space and its application to ID3 algorithm. In Proceeding of Second Int. Conf. on Bio-Inspired Computing: Theories and Applications (Zhenzhou, China, Sept. 2007), 77-81.

[13] Gehrke, J., Ganti, V. and Ramakrishnan, R. 1999. BOAT-optimistic decision tree construction. In Proceeding of the 1999 ACM SIGMOD Int. Conf. on Management of Data (Philadelphia, PA, USA, June 1999). ACM SIGMOD Record, New York, NY, 169-180. DOI=http://doi.acm.org/10.1145/304181.304197.

[14] Li, F.-C. and Li, P. 2008. The research and realization of decision trees based on credibility. In Proceeding of Int.

Conf. on Machine Learning and Cybernetics (Kunming, China, July 2008), vol. 1, 345-349.

[15] Li, Y., Li, G.-G., Li, F.-C. and Jin, C.-X. 2008. The data mining method based on second learning. In Proceeding of Int. Conf. on Machine Learning and Cybernetics (Kunming, China, July 2008), vol. 1, 340-344.

[16] Wang, Q., Wu, Y.-H., Xiao, J.-W. and Pan, G.-F. 2007. The applied research based on decision tree of data mining in third-party logistics. In Proceeding of Int. Conf. on Automation and Logistics (Aug. 2007), 1540-1544.

[17] Bensaid, A.M., Bouhouch, N., Bouhouch, R., Fellat, R. and Amri, R. 1998. Classification of ECG patterns using fuzzy rules derived from ID3-induced decision trees. In Proceeding of Conference of the North American on Fuzzy Information Processing Society – NAFIPS (Aug. 1998), 34-38.

[18] Chandra, B., Mazumdar, S., Arena, V. and Parimi, N. 2002. Elegant decision tree algorithm for classification in data mining. In Proceeding of Int. Conf. on Web Information Systems Engineering (Workshop) (Dec. 2002), 160-169.

[19] Wang, X.-Z., Yang, H.-W., Zhao, M.-H. and Sun, J. 2002. A decision tree based on hierarchical decomposition. In Proceeding of Int. Conf. on Machine Learning and Cybernetics (Now. 2002), vol. 3, 1824-1828.

[20] Quinlan, J. R. 1996. Learning decision tree classifiers. ACM Computing Surveys (CSUR) (March 1996), New York, NY, 71-72. DOI=http://doi.acm.org/10.1145/234313.234346.

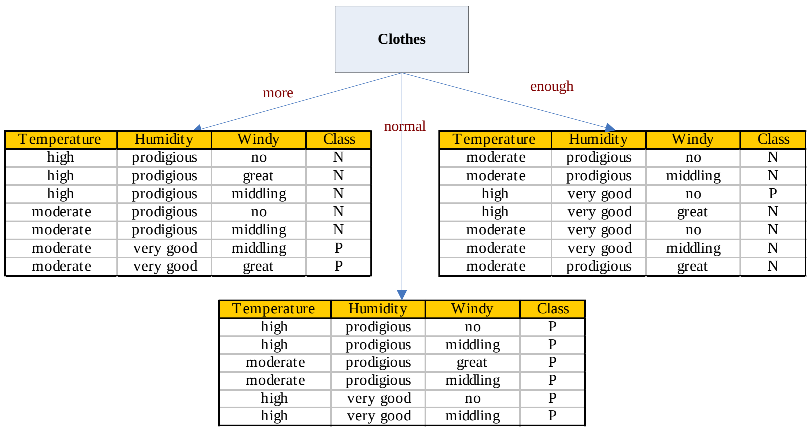

Figure 1. Branching for each sub-attribute of "clothes".

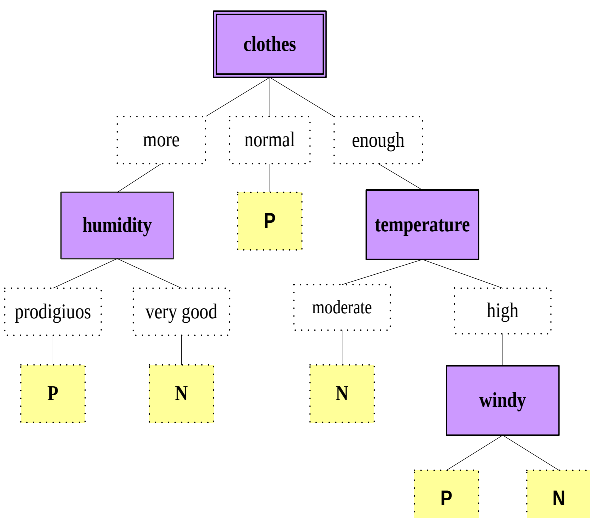

Figure 2. Final decision tree using ID3 algorithm.

Table 8. Results with respect to the attributes after changing the number of classes

Clothes				
For clothes=*"more"*	p_1=1	n_1=5	m_1=1	I(p_{11},n_{21},m_{31})=1.149
For clothes=*"normal"*	p_2=4	n_2=0	m_2=2	I(p_{12},n_{22},m_{32})=0.918
For clothes=*"enough"*	p_3=2	n_3=2	m_3=3	I(p_{13},n_{23},m_{33})=1.557
Temperature				
For temperature =*"high"*	p_1=4	n_1=4	m_1=1	I(p_{11},n_{21},m_{31})=1.392
For temperature=*"moderate"*	p_2=3	n_2=3	m_2=5	I(p_{12},n_{22},m_{32})=1.539
Windy				
For windy=*"no"*	p_1=4	n_1=3	m_1=0	I(p_{11},n_{21},m_{31})=0.985
For windy=*"great"*	p_2=2	n_2=0	m_2=3	I(p_{12},n_{22},m_{32})=0.971
For windy=*"middling"*	p_3=1	n_3=4	m_3=3	I(p_{13},n_{23},m_{33})=1.406
Humidity				
For humidity=*"prodigious"*	p_1=6	n_1=4	m_1=2	I(p_{11},n_{21},m_{31})=1.459
For humidity=*"very good"*	p_2=1	n_2=3	m_2=4	I(p_{12},n_{22},m_{32})=1.406

Attribute	Gain
clothes	0.358
temperature	0.108
windy	**0.431**
humidity	0.143