http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 12 (22): 6641-6645, 2013
ISSN 1812-5638 / DOI: 10.3923/itj.2013.6641.6645

An Improved Decision Tree Algorithm Based on the Attribute Set Dependency

[1]Yihong Cao, [1,2]Yuwan Gu, [1]Huanhuan Cai and [1]Yuqiang Sun
[1]Chang Zhou University International Institute of Ubiquitous Computing,
Jiangsu, Changzhou, 213164, China
[2]Jiang Su University College of Electronic and Information Engineering, JiangSu,
Zhenjiang, 212013, China

Abstract: The decision tree algorithm is the more popular areas of research in data mining and ID3 algorithm is the core algorithm of decision tree algorithm, through research and analysis of the ID3 algorithm, for its shortcoming of multi-value bias interrelated, difficult to remove noise and attribute is not close enough, this study presents attributes set dependence based on rough set theory, doing the attribute reduction considering properties interdependent, thereby removing redundant attributes and the algorithm of attribute set dependence is also given ,at the same time comparing complexity of the algorithm before and after improvement. The draw improved the algorithm is better than before.

Key words: Data mining, ID3 algorithm, rough set, attribute set dependence

INTRODUCTION

With the rapid development of information technology development and the data flood time, Data mining as an emerging technology arises at the historic moment. It can help users find the important value of information from huge prior unknown data, and predict the trend of the future and behavior. Decision tree is a kind of important and easy to understand method of the data mining, As the most classic of the decision tree algorithm, ID3 algorithm Hu and Wang (2008) has clear theory, simple method, learning ability, so it becomes a good example in the field of machine learning and data mining. But the ID3 algorithm also has a lot of shortcomings and insufficiency, because the ID3 algorithm is based on mutual information (information gain) of information theory, and it turns to attribute with more values. However, attribute with more values is not necessarily the best; ID3 algorithm is also more sensitive to noise, and is not easy to remove redundant data; ID3 algorithm is an unvaried decision tree, the emphasis of relationship between attribute and attribute is insufficient, this will be easy to cause the data redundancy or some attribute will be tested many times in the same path.

Recently, there have been many optimization research of decision tree on selecting test attribute strategy, such as attribute similarity put forward by literature (Hong *et al*., 2008); attribute attention put forward by literature Wang and Cao (2009) the attribute sensitivity put forward by literature Zhu and Wan (2009) etc., these strategy can avoid ID3 algorithm tending to much value, but they are all decision attribute depending on a single condition attribute, a single attribute with dependency (Li *et al*., 2011) of zero has no contribution to decision attribute and will be abandoned in reduction (Zhu *et al*., 2011). However, research shows that if the single attribute whose dependency is 0 is deleted, it often can cause discard of knowledge. So, the paper put forwards the attribute set dependency based on the theory of rough set, and ID3 algorithm based on the attribute set dependency, which can get rid of redundant data considering the mutual dependence between attributes, and can avoid multiple-valued orientation, and finally can find decision node rapidly and efficiently., So as to improve the efficiency of ID3 algorithm.

THE ID3 ALGORITHM

Quinlan Yang (2010) puts forward the ID3 algorithm in 1986. This algorithm utilizes the theory of mutual information, and selects attributes with the maximum information gain as division properties, it divides the sample set into several subsets on the basis of value of the division attribute, and each subset division also is not divided recursively, until the sample belongs to the same kind. Finally, it will return a decision tree.

Corresponding Author: Yuqiang Sun, Chang Zhou University International Institute of Ubiquitous Computing, Jiangsu, Changzhou, 213164, China

Option to split attribute standard of ID3 as follows:

Assuming T as training data set, its attribute categories are C1 and C2... , Cn, the probability division to each category of data set T is P1, P2, ,... Pn, respectively. The expectations of information needed for a given data set is:

$$H(T)=H(P_1,P_2,...,P_n)=-\sum_{i=1}^{n}p_i\ln(p_i) \quad (1)$$

Assuming dividing the data set T into T1, T2, and, ... Tn according to the conditional attribute X, to determine the amount of information needed by classification under the data set is weighted average of the information amount of each subset.

$$k=\gamma_P(Q)=\frac{caed(POS_P(Q))}{card(U)} \quad (2)$$

Information gain is that the data set T divided by condition attributes X:

$$Gain(X,T) = H(T) - H(X,T) \quad (3)$$

Disadvantage of ID3 algorithm: The decision tree generated by ID3 method is simple, clear, and can propose an easy-to-understand rule, so it is a powerful tool for gaining knowledge of data mining and machine learning. However, due to the ID3 algorithm is based on information theory of mutual information (information gain) it tends to a larger number of properties when calculating, but values with more property is not necessarily optimal; ID3 algorithm is sensitive to noise, is not easy to remove the redundant data; ID3 algorithm is an unvaried decision tree, the emphasis of relationship between attribute and attribute is insufficient, this will be easy to cause the data redundancy or some attribute will be tested many times in the same path.

THEORY OF ROUGH SET

The theory Rough Set (RS) is a data analysis theory put forward by the polish mathematician Z.P awLak (Cheng and Chen, 2011) in 1982. It can find potential hidden information from a large number of irregular data, and uses existing knowledge bas for approximation without needing any prior knowledge. And it is objective to treat problems. It has very good application in many areas, such as digital logic analysis and reduction, decision support, and machine learning. Pattern recognition, etc.

Attribute dependency based on rough set: Supposing K = (U, R) is an approximate space, P, and Q as an equivalence relation family gathers on the global U, and P, Q∈ R, if all types of Q can be defined by the types of the P, Namely the P can get the Q

Then Q depends on P, denoted by P => Q. Or that knowledge Q depends on knowledge P with dependence k (0≤k≤1) only when.

Attribute set dependence: In a decision-making system, the dependence of decision attribute to the multiple single attribute is sometimes the same, and then importance of the conditions attribute will not be able to distinguish. So, the degree of importance of the attributes of a condition not only depends on the size of single decision attribute dependence but also depends on the size of dependence of the set of attributes (two or more).

The dependence of Decision attribute Q to a single attribute is called attribute dependence, and the dependence of decision attribute Q to attribute of conditional attributes is called the set of attributes dependence. The Formula to solve the attribute set dependence remains Formula (4):

$$H(X,T)=\sum_{i=1}^{n}H(T_i)\frac{T_i}{T} \quad (4)$$

IMPROVED ALGORITHM

The thought of algorithm: Supposing information systems S = (U, A, V, f) as an ordered four Tuple, U is the domain, collection A is the collection of properties constituted by the objects, it can be divided into condition attribute set C and decision attribute set D .that is C ? D = A, C n D = Á, V is a set of attribute values, for example, Va is the range of value of attribute a, f: U×a→V is an information function, defining an information function f (x, a) Va for each a A and x V, f specify the property value of every object x in U. . Equivalence class of IND (C) is called the condition class; the equivalence class of IND (D) called the decision classes IND (C) and IND (D) are indistinguishable on behalf of C and D.

Description of algorithm: According to Eq. 4, supposing the set of condition attributes C {a, b, c ...} made by multi-attribute. Decision attribute D made by a single attribute, Solving steps of dependence of D on condition attribute set {a, b, c} as follows:

- The equivalence class division of condition attribute set {a, b, c}: if the value of an equal to the value of b, they will be divided into the same equivalence class

- The equivalence class division of decision attribute: objects with the same attribute values of attribute D will be divided into the same equivalence class
- Find the number of elements of {equivalence class of the set of condition attribute} n {equivalence class of the set of decision}, namely card ()
- $k= \gamma_C(D_s)$ = {number of elements | {equivalence class of the set of condition attribute}n {equivalence class of the set of decision}}/{numbers of Object}
- Other set of attributes Repeats steps one to four, and finally compare the size of the value k, the value of k is large and the set of attribute contains the same condition attributes, only this property is the splitting attribute, and attribute that leads to the smallest value of k is redundant attributes And it can be abandon. The rest of the property also performs steps one to four, until all attributes can be distinguished the importance

Algorithm pseudo code

```
The Function ASDM _Dtree (C: candidate attribute set, T: a training
set) returns a decision tree.
Enter the training sample set T, the condition attribute set C.
Output a decision tree.
{
1) Create a root node Root;
2) If T is made of record with the same category attribute value,
returns a single node with this value;
3) If R is empty, returns the root as a leaf node, and mark root as the
largest class of appearing in record T;
4) For each condition attribute set C in T
ASDM (C) ;//function ASDM (C) :attribute set dependence of Ci is
calculated by the formula (1);
All object sorts according to the set of attributes demanded for
solving/division of equivalence classes
nCount = 0// { equivalence class of the set of condition attribute } n{
equivalence class of the set of decision }, the initial 0
if the values of the attribute set of all objects are the same
Count = nCount + number of such objects
End for
Return nCount// total number of objects
k = nCount/the total number of decision attribute set// return the
dependence of each equivalence class
end ASDM
5) Test (which is the current test attribute of Root) = Ci with the
biggest dependence of attribute set in candidate attribute;
6) Delete Ci from R, form a new candidate attribute set C ';
7) Returns a tree, its root is tagged for T, and its branches are marked
as d1, d2, ... dm;
8) for each values ??of Test = C '
{
Grow a new child node from the node Root;
IF new leaf node corresponds to the subset of samples T 'is empty
Delete this node:
ELSE
ASDM _Dtree (C ', T' m);
}
}
```

CASE STUDY

Training set U of meteorological conditions, as shown in Table 1, we take the case of the two categories,

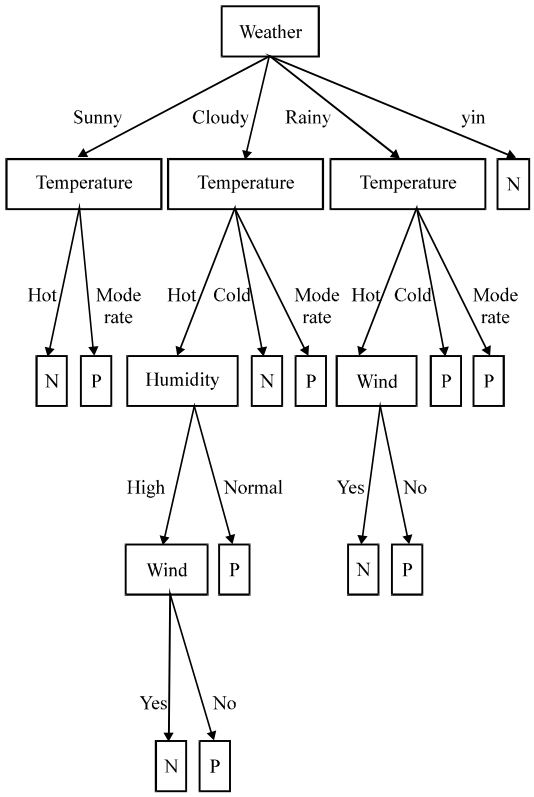

Fig. 1: ID3 decision tree

respectively, for the P and N, Class P and N class on behalf of the entity concept of positive and negative examples. Where P is suitable for outdoor, N are not suitable for outdoor.

ID3 decision tree algorithm

Solving steps of The ID3 algorithm are as follows:
Step 1: Find the expectations of the training set U
H (U) = -5/14log2 (5/14) -9/14log2 (9/14) = 0.94
Step 2: Find the information entropy of each attributes
H (weather, U) =
-4/14 * (2/4 * log2 (2/4) +2/4 * log2 (2/4)) -5/14 * (4/5 * log2 (4/5) +1/5 * log2 (1/5)) -4/14 * (2/4 * log2 (2/4) +2/4 * log2 (2/4)) +1/14 * 0 = 0.65
Similarly H (air temperature, U) = 0.69
H (humidity, U) = 0.87
H (wind, U) = 0.94
The third step: find the information gain of each attributes
Gain (weather, U) = H (U) - H (Weather, U) = 0.94-0.65 = 0.29
Similarly Gain (weather, U) = 0.25;
Gain (weather, U) = 0.07;
Gain (weather, U) = 0
The resulting weather information gain is the biggest, so chose the weather for the root node,

The eventual establishment of the ID3 decision tree shown in Fig. 1.

The depth of the tree are four layers can be seen from Fig. 1 and leaves has 12 points. The time complexity is O (nlog2 (n))

Table 1: Meteorological conditions

U	Weather	Temperature	Humidity	Wind	Category
1	sunny	hot	high	no	N
2	sunny	hot	high	no	N
3	sunny	moderate	high	no	P
4	cloudy	cold	high	no	P
5	rainy	cold	normal	no	P
6	rainy	cold	normal	yes	N
7	rainy	moderate	normal	yes	P
8	Yin	hot	high	no	N
9	cloudy	cold	normal	no	P
10	cloudy	cold	normal	no	P
11	cloudy	hot	normal	yes	P
12	cloudy	moderate	high	yes	P
13	sunny	moderate	normal	no	P
14	cloudy	cold	high	yes	N

Table 2: Simplifies table of Meteorological training set

U	a	b	c	d	e
1	0	0	0	0	0
2	0	0	0	0	0
3	0	1	0	0	1
4	1	2	0	0	1
5	2	2	1	0	1
6	2	2	1	1	0
7	2	1	1	1	1
8	3	0	0	0	0
9	2	0	1	0	1
10	1	2	1	0	1
11	1	0	1	1	1
12	1	1	0	1	1
13	0	1	1	0	1
14	1	2	0	1	0

ID3 algorithm based on attribute set dependence: For example , meteorological training set of Table 1, , the sake of brevity, we will replace the semantics of Table 1with digital code , As shown in Table 2, wherein a, b, c, d, e respectively represent the attributes of weather, temperature, humidity ,wind and categories.

Attribute a (0,1.2.3) on behalf of a (sunny, cloudy, rain, overcast), other properties are also similar.

Achievements of The improved algorithm are as following steps:

The first step: equivalence partitioning of condition attribute set {a, b}

U/{a, b, c} = {1,2} {3} {4,14} {5,6} {7} {8} {9} {10} {11} {12} {13}

U/{a, b, d} = {1,2} {3,13} {4,10} {5} {6} {7} {8} {9} {11} {12} {14}

U/{a, c, d} = {1,2,3} {4} {5,9} {6,7} {8} {10} {11} {12,14} {13}

U/{b, c, d} = {1,2,8} {3} {4} {5,10} {6} {7} {9} {11} {12} {13} {14}

Step two: equivalence class division of the decision attribute set

U/e = {1,2,6,8,14} {3,4,5,7,9,10,11,12,13}

The third step; the number of elements of {equivalence class of the set of condition attribute} n{ equivalence class of the set of decision },

For example: dependence, of D on the attribute set {a, b, c}

U/{a, b, c} n U/e = {1,2} {3} {7} {8} {9} {10} {11} {12} {13}

Card (POSC (D)) = 10

The fourth step: dependence of D on attribute set {a, b, c} is 10/14

Similarly too

Dependence of D on the attribute set {{a, b, d}is 1

Dependence of D on the attribute set {a, c, d} is 7/14

Dependence of D on the attribute set {B, C, D}is 1

It can be seen that dependence of D on the attribute set {a, b, c}, {a, b, d}, {b, c, d} is relatively higher, which contains the attribute b, so b is the most important of condition attribute. To distinguish between the importance of the other three properties, we should continue to call the above four steps. Obtained respectively dependence of D on the attribute set { a, c }, {a, d}, {c, d} are 4/14, 5/14, 4/14, thus deriving attribute c is the most unimportant , the last call of the above four steps to continue to the importance of the difference between a and d, the obtained the importance of D on the properties a and d is 1/14, 0, eventually can be drawn

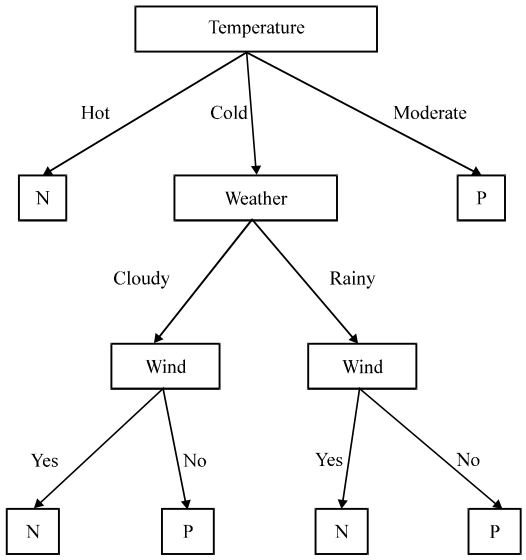

Fig. 2: Improved ID3 decision tree

to the importance of a higher than d, so finally obtained the Sort of importance of size of the attribute: b> a> d> c. and according to the user prior knowledge, it can be drawn that the general people outdoors more consider the temperature rather than the weather, and generally pay little attention to the humidity, so the humidity can not be considered, this is easier for people to be able to correct quickly determine whether it is appropriate to go out. The decision tree shown in Fig. 2.

As can be seen from Fig. 2, the depth of the tree is 3, and leaves points are a total of six. The time complexity is O (n).

Comparison of The decision trees built by the two algorithms shows that the ID3 algorithm based on attribute set dependence tree is more concise, more intuitive than theID3 algorithm built before. While selecting the decision attribute can also avoid the orientation of the multi-value, the most important thing is

that the time complexity is relatively low. They also do not take up a lot of space.

CONCLUSION

The improved ID3 algorithm based on attribute set dependence can not only get rid of redundant data Correctly but also can consider the links between attributes. Solving the best attribute to avoid the variety bias, and time complexity is relatively small. While selecting the decision attribute can also avoid the orientation of the multi-value, the time complexity is relatively lower. At the same time do not produce intermediate value, so space will also smaller

However, training for large-scale data sets, the equivalent classification efficiency is relatively low. Therefore, we should take future researching and improving.

ACKNOWLEDGMENTS

Supported by The National Natural Science Fund (11271057,51176016)and Jiangsu Province ordinary university innovative research project (CXZZ13_0691) and Natural Science Fund in JiangSu (BK2009535) and Natural Science Fund in ZheJiang (Y1100314).

REFRENCES

Cheng, M. and H.P. Chen, 2011. Based on the hadoop web log mining. Comput. Eng., 37: 37-42.

Hong, X., Z.C. Wang, Y. Liu and M.X. Yang, 2008. Linux environment frame based on PVM parallel machines. The Development and Application of Computer.

Hu, C.J. and X.W. Wang, 2008. Based on nuclear cluster system parallel programming model research. Computer Science Development No. 4.

Li, C.H., X.F. Zhang and W. Xiang, 2011. New distributed parallel computing model. Computer Engineering and Science.

Wang, H.C. and J.P. Cao, 2009. Based on SMP cluster hybrid parallel programming model research. Comput. Eng., 3: 271-273.

Yang, G.Z., 2010. HADOOP based on data mining research. Chongqing University, Chongqing, pp: 45-50.

Zhu, M., J.Y. Wan and M.W. Wang, 2011. Based on MR parallel decision tree classification algorithm design and implementation. J. Guangxi Normal Univ. (Nat. Sci. Edn.,), 29: 82-84.

Zhu, X.K. and J.Y. Wan, 2009. Parallel genetic algorithm framework research and implementation. Comput. Eng. Des., 20: 4588-4591.