Application research of decision tree algorithm in English grade analysis

Zhao Kun
Beihua University, Teacher’s college, Jilin, (CHINA)

ABSTRACT

This paper introduces and analyses the data mining in the management of students’ grades. We use the decision tree in analysis of grades and investigate attribute selection measure including data cleaning. We take course score of institute of English language for example and produce decision tree using ID3 algorithm which gives the detailed calculation process. Because the original algorithm lacks termination condition, we propose an improved algorithm which can help us to find the latency factor which impacts the grades.

KEYWORDS

Decision tree algorithm; English grade analysis; ID3 algorithm; Classification.
INTRODUCTION

With the rapid development of higher education, English grade analysis as an important guarantee for the scientific management constitutes the main part of the English educational assessment. The research on application of data mining in management of students' grades wants to talk how to get the useful uncovered information from the large amounts of data with the data mining and grade management\textsuperscript{[1-5]}. It introduces and analyses the data mining in the management of students' grades. It uses the decision tree in analysis of grades. It describes the function, status and deficiency of the management of students' grades. It tells us how to employ the decision tree in management of students' grades. It improves the ID3 arithmetic to analyze the students' grades so that we could find the latency factor which impacts the grades. If we find out the factors, we can offer the decision-making information to teachers. It also advances the quality of teaching\textsuperscript{[6-10]}. The English grade analysis helps teachers to improve the teaching quality and provides decisions for school leaders.

The decision tree-based classification model is widely used as its unique advantage. Firstly, the structure of the decision tree method is simple and it generates rules easy to understand. Secondly, the high efficiency of the decision tree model is more appropriate for the case of a large amount of data in the training set. Furthermore the computation of the decision tree algorithm is relatively not large. The decision tree method usually does not require knowledge of the training data, and specializes in the treatment of non-numeric data. Finally, the decision tree method has high classification accuracy, and it is to identify common characteristics of library objects, and classify them in accordance with the classification model. The original decision tree algorithm uses the top-down recursive way\textsuperscript{[11-12]}. Comparison of property values is done in the internal nodes of the decision tree and according to the different property values judge down branches from the node. We get conclusion from the decision tree leaf node. Therefore, a path from the root to the leaf node corresponds to a set of disjunctive expressions rules. The decision tree generation algorithm is divided into two steps\textsuperscript{[13-15]}. The first step is the generation of the tree, and at the beginning all the data is in the root node, then do the recursive data slice. Tree pruning is to remove some of the noise or abnormal data. Conditions of decision tree to stop splitting is that a node data belongs to the same category and there are not attributes used to split the data.

In the next section, we introduce construction of decision tree. In Section 3 we introduce attribute selection measure. In Section 4, we do empirical research based on ID3 algorithm and propose an improved algorithm. In Section 5 we conclude the paper and give some remarks.

CONSTRUCTION OF DECISION TREE USING ID3

The growing step of the decision tree is shown in Figure 1. Decision tree generation algorithm is described as follows. The name of the algorithm is $\text{Generate decision tree}$ which produce a decision tree by given training data. The input is training samples which is represented with discrete values. Candidate attribute set is attribute. The output is a decision tree.

Step1. Set up node N. If samples is in a same class C then return N as lead node and label it with C.

Step2. If attribute list is empty, then return N as leaf node and label it with the most common class in the samples.

Step3. Choose $\text{test_attribute}$ with information gain in the attribute list, and label N as $\text{test_attribute}$.

Step4. While each $a_i$ in every $\text{test_attribute}$ do the following operation.

Step5. Node N produces a branch which meets the condition of $\text{test_attribute} = a_i$.

Step6. Suppose $s_i$ is sample set of $\text{test_attribute} = a_i$ in the samples. If $s_i$ is empty, then plus a leaf and label it as the most common class. Otherwise plus a node which was returned by $\text{Generate decision tree}(s, \text{attribute list} - \text{test_attribute})$. 


AN IMPROVED ALGORITHM

Attribute selection measure
Suppose $S$ is data sample set of $s$ number and class label attribute has $m$ different values $C_i (i=1,2,\cdots,m)$. Suppose $S_i$ is the number of sample of class $C_i$ in $S$. For a given sample classification the demanded expectation information is given by formula 1 [11-12].

$$I(s_{ij},s_{2j},K,s_{mj}) = -\sum_{i=1}^{m} p_{ij} \log_2 p_{ij} (i=1,2,\cdots,K,n)$$

(1)

$$E(A) = \sum_{j=1}^{V} \left( \frac{S_{1j} + S_{2j} + \cdots + S_{mj}}{S} \right) I(S_{1j},S_{2j},K,S_{mj})$$

(2)

$p_i$ is probability that random sample belongs to $C_i$ and is estimated by $s_i/s$. Suppose attribute $A$ has $V$ different values $(a_1,a_2,\cdots,a_v)$. We can use attribute $A$ to classify $S$ into $V$ number of subset $(S_1,S_2,\cdots,S_V)$. Suppose $S_i$ is the number of class $C_i$ in subset $S_j$. The expected information of subset is shown in formula 2. $\left( \frac{S_{1j} + S_{2j} + \cdots + S_{mj}}{S} \right)$ is the weight of the $j$-th subset. For a given subset $S_j$ formula 3 sets up [13].

$$I(s_{ij},s_{2j},K,s_{mj}) = -\sum_{i=1}^{m} p_{ij} \log_2 p_{ij} (i=1,2,\cdots,K,n)$$

(3)
\[ p_{ij} = \frac{s_j}{s_i} \] is the probability that samples of \( s_j \) belongs to class \( C_i \). If we branch in \( A \), the information gain is shown in formula 4\(^{[14]}\).

\[
Gain(A) = I(s_1, s_2, \cdots, s_m) - E(A)
\]

(4)

**The improved algorithm**

The improved algorithm is as follows. Function \( \text{Generate\_decision\_tree} \) (training samples, candidate attribute attribute_list)

\[
\{ \text{Set up node } N; \\
\text{If samples are in the same class } C \text{ then} \\
\text{Return } N \text{ as leaf node and label it with } C; \\
\text{Record statistical data meeting the conditions on the leaf node;} \\
\text{If attribute\_list is empty then} \\
\text{Return } N \text{ as the leaf node and label it as the most common class of samples;} \\
\text{Record statistical data meeting the conditions on the leaf node;} \\
\text{Suppose } \text{GainMax} = \max \left( \text{Gain1}, \text{Gain2}, \cdots, \text{Gainn} \right) \\
\text{If } \text{GainMax} < \text{threshold} \\
\text{Return } N \text{ as the leaf node and label it as the most common class of samples;} \\
\text{Choose attribute with the highest information gain of attribute\_list;} \\
\text{Label } N \text{ as test\_attribute;} \\
\text{For each } a_i \text{ of test\_attribute, produce a branch from node } N \text{ meeting the condition of } \\
\text{test\_attribute} = a_i ; \\
\text{Suppose } s_i \text{ sample set of samples meeting the condition of test\_attribute} = a_i ; \\
\text{If } s_i \text{ is empty then Record statistical data meeting the conditions on the leaf node;Add a leaf and label it as the most common class of samples;} \\
\text{Else add a node returned by } \text{Generate\_decision\_tree} \left( s_i, \text{attribute\_list\_test\_attribute} \right); \\
\}

**EMPIRICAL RESEARCH**

**Data cleaning**

This paper takes course score of institute of English language for example. Examination score of the students is shown in TABLE 1.

**TABLE 1: Examination score of the students**

<table>
<thead>
<tr>
<th>Course code</th>
<th>Whether re-learning</th>
<th>Paper difficulty</th>
<th>Whether required course</th>
<th>score</th>
</tr>
</thead>
<tbody>
<tr>
<td>110101290</td>
<td>no</td>
<td>high</td>
<td>yes</td>
<td>89</td>
</tr>
<tr>
<td>H200104088</td>
<td>no</td>
<td>middle</td>
<td>yes</td>
<td>75</td>
</tr>
<tr>
<td>H200116090</td>
<td>yes</td>
<td>middle</td>
<td>no</td>
<td>80</td>
</tr>
<tr>
<td>H120101160</td>
<td>yes</td>
<td>high</td>
<td>yes</td>
<td>65</td>
</tr>
<tr>
<td>120101288</td>
<td>yes</td>
<td>middle</td>
<td>yes</td>
<td>70</td>
</tr>
<tr>
<td>H200152069</td>
<td>no</td>
<td>low</td>
<td>no</td>
<td>90</td>
</tr>
</tbody>
</table>

Data in TABLE 1 is not suitable for classification, so we firstly do data cleaning. According to the general course, basic course, professional basic course and specialized course, classify the course into A, B, C, D. Score is divided into three categories outstanding, medium, general. Paper difficulty is divided into three categories 1, 2, 3. Such as
Update ks set ci_pi='outstanding' where ci_pj>='85'
Update ks set ci_pi='medium' where ci_pj>='75' and ci_pj<>'85'
Update ks set ci_pi='general' where ci_pj>='60' and ci_pj<>'75'
Update ks set sjnd='high' where sjnd='1'
Update ks set sjnd='medium' where sjnd='2'
Update ks set sjnd='low' where sjnd='3'

Result of ID3 algorithm

TABLE 2 is training set of student test scores situation information after data cleaning. We classify the samples into three categories. $C_1 =$ "outstanding", $C_2 =$ "medium", $C_3 =$ "general", $s_1 = 300$, $s_2 = 1950$, $s_3 = 880$, $s = 3130$. According to formula 1, we obtain $I(s_1, s_2, s_3) = (300, 1950, 880) = -(300 / 3130) / \log_2 (300 / 3130)$.

\[-(1950 / 3130) \log_2 (1950 / 3130) - (880 / 3130) \log_2 (880 / 3130)\]
\[= 1.256003.\]

Entropy of every attribute is calculated as follows. Firstly calculate whether re-learning. For yes, $s_{11} = 210$, $s_{21} = 950$, $s_{31} = 580$.

$I(s_{11}, s_{21}, s_{31}) = (210, 950, 580)$
\[= -(210 / 1740) \log_2 (210 / 1740) - (950 / 1740) \log_2 (950 / 1740)\]
\[= -(580 / 1740) \log_2 (580 / 1740) = 1.074901.\]

For no, $s_{12} = 90$, $s_{22} = 1000$, $s_{32} = 300$.

$I(s_{12}, s_{22}, s_{32}) = (90, 1000, 300)$
\[= -(90 / 1390) \log_2 (90 / 1390) - (1000 / 1390) \log_2 (1000 / 1390)\]
\[= -(300 / 1390) \log_2 (300 / 1390) = 1.373186.\]

IF samples are classified according to whether re-learning, the expected information is

$E("\text{whether re-learning}") = (1740 / 3130) \cdot I(s_{11}, s_{21}, s_{31}) + (1390 / 3130) \cdot I(s_{12}, s_{22}, s_{32})$
\[= 0.555911 \cdot 1.074901 + 0.444089 \cdot 1.373186 = 1.240721.\] So the information gain is

$Gain("\text{whether re-learning}") = I(s_1, s_2, s_3) - E("\text{whether re-learning}") = 0.015282.$

Secondly calculate course type, when it is A, $s_{11} = 110, s_{21} = 200, s_{31} = 580$.

$I(s_{11}, s_{21}, s_{31}) = (110, 200, 580)$
\[= -(110 / 890) \log_2 (110 / 890) - (200 / 890) \log_2 (200 / 890) - (580 / 890) \log_2 (580 / 890)\]
\[= 1.259382.\]

For course type B, $s_{12} = 100, s_{22} = 400, s_{32} = 0$. 

\[-(1950 / 3130) \log_2 (1950 / 3130) - (880 / 3130) \log_2 (880 / 3130)\]
\[= 1.256003.\]
\( I(s_{12}, s_{22}, s_{32}) = (100, 400, 0) \)
\[ = -(100 / 500) \log_2 (100 / 500) - (400 / 500) \log_2 (400 / 500) - 0 \]
\[ = 0.721928. \]

For course type C, \( s_{13} = 0, s_{23} = 550, s_{33} = 0. \)
\[ I(s_{13}, s_{23}, s_{33}) = (0, 550, 0) \]
\[ = -(0 / 550) \log_2 (0 / 550) - (550 / 550) \log_2 (550 / 550) - 0 \]
\[ = 1.168009. \]

For course type D, \( s_{14} = 90, s_{24} = 800, s_{34} = 300. \)
\[ I(s_{14}, s_{24}, s_{34}) = (90, 800, 300) \]
\[ = -(90 / 1190) \log_2 (90 / 1190) - (800 / 1190) \log_2 (800 / 1190) - (300 / 1190) \log_2 (300 / 1190) \]
\[ = 1.168009. \]

\[ E("course type") = (890 / 3130) \cdot I(s_{11}, s_{21}, s_{31}) + (500 / 3130) \cdot I(s_{12}, s_{22}, s_{32}) \]
\[ + (550 / 3130) \cdot I(s_{13}, s_{23}, s_{33}) + (1190 / 3130) \cdot I(s_{14}, s_{24}, s_{34}) \]
\[ = 0.91749. \]

\[ Gain("course type") = 1.256003 - 0.917949 = 0.338513. \]

Thirdly calculate paper difficulty. For high, \( s_{11} = 110, s_{21} = 900, s_{31} = 280. \)
\[ I(s_{11}, s_{21}, s_{31}) = (110, 900, 280) \]
\[ = -(110 / 1290) \log_2 (110 / 1290) - (900 / 1290) \log_2 (900 / 1290) - (280 / 1290) \log_2 (280 / 1290) \]
\[ = 1.14385. \]

For medium, \( s_{12} = 190, s_{22} = 700, s_{32} = 300. \)
\[ I(s_{12}, s_{22}, s_{32}) = (190, 700, 300) \]
\[ = -(190 / 1190) \log_2 (190 / 1190) - (700 / 1190) \log_2 (700 / 1190) - (300 / 1190) \log_2 (300 / 1190) \]
\[ = 1.374086. \]

For low, \( s_{13} = 0, s_{23} = 350, s_{33} = 300. \)
\[ I(s_{13}, s_{23}, s_{33}) = (0, 350, 300) \]
\[ = -(0 / 650) \log_2 (0 / 650) - (350 / 650) \log_2 (350 / 650) - (300 / 650) \log_2 (300 / 650) = 0.995727. \]

\[ E("paper difficulty") = (1290 / 3130) \cdot I(s_{11}, s_{21}, s_{31}) + (1190 / 3130) \cdot I(s_{12}, s_{22}, s_{32}) \]
\[ + (650 / 3130) \cdot I(s_{13}, s_{23}, s_{33}) = 1.200512. \]

\[ Gain("paper difficulty") = 1.256003 - 1.200512 = 0.55497. \]

Fourthly calculate whether required course. For yes, \( s_{11} = 210, s_{21} = 850, s_{31} = 600 \)
\[ I(s_{11}, s_{21}, s_{31}) = (210, 850, 600) = -\frac{210}{1660} \log_2(\frac{210}{1660}) - \frac{850}{1660} \log_2(\frac{850}{1660}) - \frac{600}{1660} \log_2(\frac{600}{1660}) = 1.220681. \]

For no, \( s_{12} = 90, s_{22} = 1100, s_{32} = 280 \)

\[ I(s_{12}, s_{22}, s_{32}) = (90, 1100, 280) = -\frac{90}{1470} \log_2(\frac{90}{1470}) - \frac{1100}{1470} \log_2(\frac{1100}{1470}) - \frac{280}{1470} \log_2(\frac{280}{1470}) = 1.015442. \]

\[ E("\text{whether required}\") = (1660 / 3130) \cdot I(s_{11}, s_{21}, s_{31}) + (1470 / 3130) \cdot I(s_{12}, s_{22}, s_{32}) = 1.220681. \]

\[ \text{Gain("\text{whether required}\")} = 1.256003 - 1.220681 = 0.035322. \]

**TABLE 2: Training set of student test scores**

<table>
<thead>
<tr>
<th>Course type</th>
<th>Whether re-learning</th>
<th>Paper difficulty</th>
<th>Whether required</th>
<th>Score</th>
<th>Statistical data</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>no</td>
<td>medium</td>
<td>no</td>
<td>90</td>
<td>outstanding</td>
</tr>
<tr>
<td>B</td>
<td>yes</td>
<td>medium</td>
<td>yes</td>
<td>100</td>
<td>outstanding</td>
</tr>
<tr>
<td>A</td>
<td>yes</td>
<td>high</td>
<td>yes</td>
<td>200</td>
<td>medium</td>
</tr>
<tr>
<td>D</td>
<td>no</td>
<td>low</td>
<td>no</td>
<td>350</td>
<td>medium</td>
</tr>
<tr>
<td>C</td>
<td>yes</td>
<td>medium</td>
<td>yes</td>
<td>300</td>
<td>general</td>
</tr>
<tr>
<td>A</td>
<td>yes</td>
<td>high</td>
<td>no</td>
<td>250</td>
<td>medium</td>
</tr>
<tr>
<td>B</td>
<td>no</td>
<td>high</td>
<td>no</td>
<td>300</td>
<td>medium</td>
</tr>
<tr>
<td>A</td>
<td>yes</td>
<td>high</td>
<td>yes</td>
<td>110</td>
<td>outstanding</td>
</tr>
<tr>
<td>D</td>
<td>yes</td>
<td>medium</td>
<td>yes</td>
<td>500</td>
<td>medium</td>
</tr>
<tr>
<td>D</td>
<td>no</td>
<td>low</td>
<td>yes</td>
<td>300</td>
<td>general</td>
</tr>
<tr>
<td>A</td>
<td>yes</td>
<td>high</td>
<td>no</td>
<td>280</td>
<td>general</td>
</tr>
<tr>
<td>B</td>
<td>no</td>
<td>high</td>
<td>yes</td>
<td>150</td>
<td>medium</td>
</tr>
<tr>
<td>C</td>
<td>no</td>
<td>medium</td>
<td>no</td>
<td>200</td>
<td>medium</td>
</tr>
</tbody>
</table>

**Result of improved algorithm**

The original algorithm lacks termination condition. There are only two records for a subtree to be classified which is shown in **TABLE 3**.

**TABLE 3: Special case for classification of the subtree**

<table>
<thead>
<tr>
<th>Course type</th>
<th>Whether re-learning</th>
<th>Paper difficulty</th>
<th>Whether required</th>
<th>Score</th>
<th>Statistical data</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>no</td>
<td>high</td>
<td>yes</td>
<td>medium</td>
<td>15</td>
</tr>
<tr>
<td>A</td>
<td>no</td>
<td>high</td>
<td>yes</td>
<td>general</td>
<td>20</td>
</tr>
</tbody>
</table>
All Gains calculated are 0.00, and GainMax=0.00 which does not conform to recursive termination condition of the original algorithm in TABLE 3. The tree obtained is not reasonable, so we adopt the improved algorithm and decision tree using improved algorithm is shown in Figure 2.

CONCLUSIONS

In this paper we study construction of decision tree and attribute selection measure. Because the original algorithm lacks termination condition, we propose an improved algorithm. We take course score of institute of English language for example and we could find the latency factor which impacts the grades.

REFERENCES