Comparison of ID3 and Its Generalized Version

Tzung-Pei Hong
Department of Computer Science
Chung-Hua Polytechnic Institute
Hsin-Chu, 30067, Taiwan, R.O.C.

Shian-Shyong Tseng
Department of Computer and Information Science
National Chiao-Tung University
Hsin-Chu, 30050, Taiwan, R.O.C.

Abstract

A generalized ID3 learning algorithm, which possesses more processing capabilities than the original ID3, has been proposed here. This generalized ID3 learning algorithm can manage uncertain training instances and consider different importance of different training instances in finding an appropriate decision tree in a noisy environment. If appropriate priori domain knowledge is available, this algorithm can further utilize it in reducing the effect of noise. Relations on generality, accuracy, tree complexity and time complexity of ID3 and generalized ID3 are also discussed, with experiments verifying their correctness.

1 Introduction

Many factors exist in designing a good learning system. Among them, reducing the influence of noise is quite important. Validity and relevance of the finally learned concepts heavily depends on the accuracy of chosen training instances. In real applications, the data provided to the learning systems by experts, teachers, or users usually contain noise. Noise can be expected to greatly affect the formation and use of the finally learned concepts[6].

Modifying traditional learning methods for working well in noisy environments is then very important. Several extensions based on ID3 were proposed for managing noisy training instances[1][4][7][8]. They, however, seldom discuss the situations where inconclusive training instances may exist, different training instances may have different importance, different classes may have different noise ratios, and priori preferred hypotheses may be available. In this paper, generalized ID3 learning algorithms have then been presented for taking care of the above additional capabilities.

2 Review of ID3 learning algorithm

In 1983, Quinlan proposed a learning algorithm ID3[5] which tries to form a decision tree from a set of training instances. ID3 uses the heuristics of minimizing the "entropy" in determining which attribute should be next selected in the decision tree. If Attribute A has m values (i.e., A1, A2, . . . , Am) and the training subset Si having attribute value Ai can be partitioned into n+ positive training instances and n− negative training instances, then the entropy (E) by choosing A as the next attribute will be calculated by the following formula:

\[
E = \sum_{i=0}^{m} -\frac{n^+_i}{n_i+n^-} \log_2 \frac{n^+_i}{n_i+n^-} - \frac{n^-_i}{n_i+n^+} \log_2 \frac{n^-_i}{n_i+n^+}
\]

Among all the feasible attributes, the one which causes the minimum entropy will be chosen as the next attribute. The same procedure is then repeated until each terminal node in the decision tree contains only training instances of the same class.

3 Generalized ID3 learning algorithm

In this section, ID3 has been generalized for learning concepts well under the mentioned constraints.

3.1 Managing training instances with uncertainty and representative

Managing training instances with uncertainty and representative is very easy and natural for ID3. An uncertain training instance can be thought of as partially positive and partially negative. A training instance with uncertainty u (0 to 1, 0 represents a completely negative instance and 1 represents a completely positive instance) and degree of representative
If \( r (0 \leq r \leq 1) \) then has a consequence, measured as \( u \times r \), on inducing the positive class and has a consequence, measured as \( (1 - u) \times r \), on inducing the negative class[2][3].

Assume a training subset includes \( n \) training instances, each with uncertainty \( u_i \) and representative \( r_i \) \((1 \leq i \leq n, i \) is an integer)\. The generalized ID3 learning algorithm will calculate its entropy as:

\[
E = -n^+ \log_2 \frac{n^+}{n^+ + n^-} - n^- \log_2 \frac{n^-}{n^+ + n^-} = -(\sum_{i=1}^{n} r_i u_i) \log_2 \frac{\sum_{i=1}^{n} r_i u_i}{\sum_{i=1}^{n} r_i} - (\sum_{i=1}^{n} r_i (1 - u_i)) \log_2 \frac{\sum_{i=1}^{n} r_i (1 - u_i)}{\sum_{i=1}^{n} r_i} = (\sum_{i=1}^{n} r_i) (-P^+ \log_2 P^+ - P^- \log_2 P^-)
\]

The generalized ID3 learning algorithm then chooses the attribute with the minimum entropy as the next one to use.

### 3.2 Managing noise

A pruning method which does not have to first generate a full decision trees is desired since uncertain training instances may exist. A learning method called "Assistant"[8] can fortunately satisfy the above requirement. Assistant terminates the growth of a node if its ratio of training instances of the most common class is equal to or larger than a predefined threshold. Assistant can obviously do the pruning work well without first generating a full decision tree. It, however, decides the entropy of a node without considering whether its children satisfy the termination criterion.

A generalized ID3 learning algorithm will be proposed later for taking care of this situation.

### 3.3 Managing training classes with different noise ratios

Training instances of different classes may sometimes have different noise ratios. For example, given positive training instances may be noise-free and given negative training instances may contain 10% noise. A solution is that two termination criteria \((TE_p \text{ and } TE_n)\) may simultaneously exist, one for the positive class and the other for the negative class.

Values of \( TE_p \text{ and } TE_n \) can be determined by respectively estimating noise ratios in the positive class and the negative class. As a principle, a lower noise ratio of positive (negative) training instances will cause a higher value of \( TE_n \) \((TE_p)\).

### 3.4 Utilizing priori domain knowledge in reducing effect of noise

Some hypotheses derived from other training instances, from other available sources, or given by other experts, should still have more or less effect in inducing the final result. Different priori hypotheses from different sources may then have different validity. The higher the validity of a priori hypothesis is, the more important this hypothesis is. The generalized learning algorithm can then be stated as:

The generalized ID3 learning algorithm:

**INPUT:** A set of training instances \( S' \), a set of attributes \( A \), a set of hypotheses \( H \), and two termination criteria \( TE_p \) and \( TE_n \) \((TE_p \text{ and } TE_n \) may be the same).

**OUTPUT:** An appropriate decision tree.

**STEP 1.** For each priori hypothesis \( h \) with validity \( v_j \), generate the set of instances \( T' \) covered by \( h \) if \( h \) is for the positive class, each generated instance is then attached with uncertainty \( u = 1 \) and representative \( r = v_j \); if \( h \) is for the negative class, each generated instance is then attached with \( u = 0 \) and \( r = v_j \).

**STEP 2.** Combine all generated training instances with the given training set \( S' \) into a new training set \( S \).

**STEP 3.** If the ratio of training instances of the positive(negative) class is equal to or larger than \( TE_p(TE_n) \), designate the current node as a leaf node in the decision tree, labelled with the positive(negative) class. Otherwise, go to **STEP 4**.

**STEP 4.** If the number of the remaining candidate attributes is 0, designate the current node as a leaf node for the tree, labelled with the most common class. Otherwise, go to **STEP 5**.

**STEP 5.** Let \( A = \{A_1, A_2, \ldots, A_k\} \) be the set of remaining candidate attributes. For each \( A_i \) \((1 \leq i \leq k)\), assume the possible values of \( A_i \) are in \( \{A_{i,1}, A_{i,2}, \ldots, A_{i,m}\} \). Use the attribute values of \( A_i \) to partition \( S \) into the mutually exclusive and exhaustive subsets \( \{S_{i,1}, S_{i,2}, \ldots, S_{i,m}\} \). If the ratio...
of training instances of the most common class in 
\( S_{i,j} \) \((1 \leq j \leq v)\) is equal to or larger than \( T E_p \), then Entropy \( E_{i,j} = (\sum_{k=1}^{n_i} r_k)(-T E_p \log_2 T E_p - (1 - T E_p) \log_2 (1 - T E_p)) \); if the ratio of training instances of the most common class in \( S_{i,j} \) is equal to or larger than \( T E_n \), then Entropy \( E_{i,j} = (\sum_{k=1}^{n_i} r_k)(-T E_n \log_2 T E_n - (1 - T E_n) \log_2 (1 - T E_n)) \); otherwise, \( E_{i,j} = (\sum_{k=1}^{n_i} r_k)(-P_i^+ \log_2 P_i^+ - P_i^- \log_2 P_i^-) \). Calculate the Entropy of \( A_i \) as \( E_i = \sum_{j=1}^{v} E_{i,j} \).

**STEP 6.** Choose the attribute with the minimum entropy as the best attribute. Let \( A_{best} \) be the attribute that has been selected as the current node.

**STEP 7.** Use the attribute values of \( A_{best} \) to partition \( S \) into the mutually exclusive and exhaustive subsets \( \{S_{best,1}, S_{best,2}, \ldots, S_{best,v}\} \). Each subset \( S_{best,j} \) contains those examples in \( S \) that have value \( A_{best,j} \) for attribute \( A_{best} \).

**STEP 8.** Create a child node in the tree for each attribute value \( A_{best,j} \) and corresponding subset \( S_{best,j} \). Label the arc from the current node to the child node with the attribute value \( A_{best,j} \).

**STEP 9.** Recursively call the procedure on the subset \( S_{best,j} \) with the set of available attributes \( A-a_{best} \) for each child node.

### 4 Relations between ID3 and generalized ID3

In this section, relations between ID3 and generalized ID3 have been discussed according to the following consideration:

1. generality,
2. accuracy of classifying training data and testing data,
3. complexity of decision tree, and
4. time complexity.

In generality, ID3 can be thought of as a special case of generalized ID3 by assigning the uncertainty of all positive training instances as \( 1 \), the uncertainty of all negative training instance as \( 0 \), the representative of all training instances as the same value, the termination criteria \( T E_p \) and \( T E_n \) as 100%, and no available priori domain knowledge. The following relation of generality then arrives:

\[
\text{Gen(ID3)} \leq \text{Gen(generalized ID3)}
\]

In accuracy of classifying training data, ID3 apparently has the higher accuracy since no pruning is done. In accuracy of classifying testing data, the better algorithm is, however, dependant on application domains. For the application domains where reliable domain knowledge is available and noise exists, relation about accuracy of classifying testing data is expected as:

\[
\text{Accuracy(ID3)} \leq \text{Accuracy(generalized ID3)}
\]

In complexity of decision tree, the relation is undeterministic, dependant on the available priori domain knowledge, noise, and the termination criteria. If the available priori domain knowledge is accurate and the termination criteria are appropriately chosen, the generated decision tree (in the noisy environment) by generalized ID3 will then be closer to the desired concepts than ID3, thus being more concise. The following relation is then expected:

\[
\text{Tree(ID3)} \geq \text{Tree(generalized ID3)}
\]

At last, in time complexity, ID3 must generate a full decision tree without being pruned; generalized ID3, however, usually generates pruned decision trees although it must compare the ratio of training instances of the most common class with \( T E \). Since time complexity heavily depends on the tree complexity, the following relation is then expected under the accurate priori domain knowledge and appropriate termination criteria (although generalized ID3 processes more training instances then ID3):

\[
\text{Time(ID3)} \geq \text{Time(generalized ID3)}
\]

### 5 Experimental analysis

Experiments implemented by C language at Sun 4/IPC are made in this section for demonstrating validity of the above relations. There are totally eight attributes, with their numbers of possible values being randomly generated as (Table 1): Instances are generated according to the following rules:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
<th>g</th>
<th>h</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Possible Values</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td>6</td>
<td>3</td>
<td>7</td>
<td>5</td>
<td>2</td>
</tr>
</tbody>
</table>
Rule 1: $a_1 \land c_2 \rightarrow \text{class 1}$,
Rule 2: $b_2 \rightarrow \text{class 1}$,
Rule 3: $f_5 \rightarrow \text{class 1}$,
Rule 4: $g_7 \rightarrow \text{class 1}$,
Rule 5: $e_3 \land d_6 \rightarrow \text{class 1}$,
Rule 6: all others are of class 2.

19440 instances of Class 1 and 18360 instances of Class 2 are then generated. 80% of generated instances are taken as training data and the others are taken as testing data. Noise is then added to the training data by converting 15% of positive training instances to negative ones and converting 5% of negative training instances to positive ones. Assume Rule 1 and Rule 4 have been known as the priori domain knowledge with validity=0.8. Accuracy, numbers of nodes, and execution time of original ID3 and generalized ID3 ($TE_p = 85\%, TE_n = 95\%$) are shown in Table 2.

Table 2: Comparison of ID3 and Generalized ID3

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>ID3</th>
<th>Generalized ID3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>0.820</td>
<td>0.962</td>
</tr>
<tr>
<td>Number of Nodes</td>
<td>17715</td>
<td>4935</td>
</tr>
<tr>
<td>Time (seconds)</td>
<td>12.680</td>
<td>8.610</td>
</tr>
</tbody>
</table>

It is easily seen that Generalized ID3 performs much better than original ID3.

6 Conclusions and future work

We have proposed a generalized ID3 learning algorithm, which possesses more processing capabilities than the original ID3. Theoretical relations on generality, accuracy, tree complexity and time complexity are also discussed, with experiments verifying their correctness. The problem with the generalized ID3 learning algorithm lies in that when no accurate statistical information is available, the applied parameters are often subjectively assigned by experts or users. If these parameters are not appropriately assigned, the derived decision tree will then have a poor accuracy in classifying future data. Developing suitable models for correctly finding the values of parameters is then an important open problem. There is much work to be done in this important area.

Acknowledgements

This research was supported by the National Science Council of the Republic of China under NSC81-0408-E009-16. We would also like to thank Miss J. M. Wei for her assistance in making these experiments.

References