Induction of Decision Trees from Inconclusive Data

Scott Spangler
Knowledge Engineering Group
Advanced Engineering Staff
GM Technical Center
Warren, MI 48090, USA
Spangler@gmr.com

Usama M. Fayyad
Artificial Intelligence Lab
EECS Department
The University of Michigan
Ann Arbor, MI 48109, USA
Fayyad@ub.cc.umich.edu

Ramasamy Uthurusamy
Computer Science Department
GM Research Laboratories
GM Technical Center
Warren, MI 48090, USA
Samy@gmr.com

Abstract: Learning from *inconclusive data* is an important problem that has not been addressed in the concept learning literature. In this paper, we define inconclusiveness and illustrate why ID3-like algorithms are bound to result in overspecialized classifications when trained on inconclusive data. We address the difficult problem of deciding when to stop specialization during top-down decision tree generation, and describe a modified version of Quinlan's ID3 algorithm, called INFERULE, which addresses some of the problems involved in learning from inconclusive data. Results show that INFERULE outperformed ID3 (with and without pruning) in tests on a real-world diagnostic database containing automobile repair cases.

1 Introduction

One of the most important problems standing in the way of applying machine induction algorithms, such as ID3[Quin86a] and its variants, to real-world data has not been addressed in the machine learning literature. This deficiency in ID3-like programs arises from the implicit assumption that enough information is available in the data to decide exactly how each data point should be classified. The INFERULE algorithm described in this paper is designed primarily to address this problem of *inconclusive* data sets. This problem arises when the attributes used in describing a set of examples are not sufficient to specify exactly one outcome (class) per example. In such a situation, given a training set of examples, and sufficiently many (possibly irrelevant) attributes, it may be possible to classify the training set perfectly. However, such a classification will necessarily fail on future test sets of unseen examples. First, we define what we mean by *inconclusive data.*

Definition: A data set of examples expressed in terms of a set of attributes that are known to be noise-free is said to be *inconclusive* if there exists no set of rules utilizing only the given attributes that classifies *all possible (observed and unobserved)* examples perfectly. i.e. the available attributes do not contain the necessary information for predicting a unique outcome for each example.

Note that it is possible for the program to classify the *training set* perfectly even when the data is inconclusive. The above definition suggests that if some set of absolutely correct rules exists, then it must necessarily make use of some attribute(s) not provided to the learning program. We call such a set *the governing rule set.* Thus the governing rule set is not discoverable by a learning program that is given an inconclusive training set. A sure sign of inconclusiveness is that, given data that is known to have no (or very little) noise, the learning program always produces a perfect classification of the training set that has a *high error rate* when used to classify new (test) data.

The only proper way to handle inconclusive data is to resort to *probabilistic rather than categorical classification* rules. In this case a rule may predict more than one outcome. Associated with each outcome is a likelihood measure (probability)[Brei84,Quin86a].

In section 2.2 we discuss how inconclusive data differs from noisy data as discussed in the machine learning literature. To date, induction research has not really addressed the problems arising from inconclusive data. Unfortunately, such data sets are not rare in industry. In fact, it has been our experience that most diagnostic databases commonly found in an industrial environment contain only a fraction of the information necessary to correctly classify unseen cases.

Yet such data sets, though imperfect, are certainly far from useless. There may be many important and useful rules in the governing rule set that do not reference inaccessible attributes. In addition it is nearly always helpful in diagnosis to narrow down the possible classifications to one of a few possibilities as opposed to hundreds. Our experience with large inconclusive data sets at General Motors, led us to investigate the possibility of a learning algorithm that can extract useful information. Our application domain is a very large database of car problems and their fixes. Using ID3 and some of its variants to induce rule sets from training subsets of this database always resulted in a perfect classifier for the particular training sets. However, the classifier had very low accuracy on predicting the outcomes of unseen cases even with very large training sets.

Case	Model	Temp-erature	Day	Customer Abuse	Outcome: Replace
1	Basic	High	M	Dropped	Casing
2	Ultima	Low	Tu	Dropped	Casing
3	Excel	Low	W	Dropped	Casing
4	Excel	High	W	Dropped	Widget
5	Basic	Low	Th	Dropped	Widget
6	Ultima	Low	F	Dropped	Casing
7	Basic	Low	M	Kicked	woozle
8	Ultima	Low	M	Kicked	Nozzle

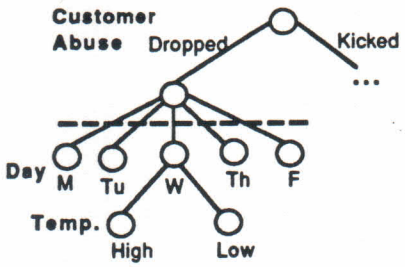

Figure 1: Data Set for Diagnosing Widget Failures and the Corresponding ID3 Tree

2 Shortcomings of ID3

ID3[Quin86a] induces a decision tree for classifying the examples in the training set. The decision tree is generated by setting the root node to be the set of training examples, partitioning the examples in the root node thus creating child nodes, and recursively partitioning each of the child nodes to get the next levels in the tree. Thus for each (non-leaf) node, ID3 selects an attribute, and creates a branch *for each value* of that attribute appearing in the subset of examples corresponding to the node. This process of specialization is then applied recursively to each non-leaf node. A set of examples along with the decision tree generated by ID3 are shown in figure 1.

The formulation of an algorithm for decision-tree generation requires the specification of the following 4 rules[Brei84]:

Rule 1: selects an attribute to use in partitioning the examples at a node.
Rule 2: chooses a particular partition of the examples based on the attribute chosen by rule 1.
Rule 3: decides when to stop partitioning a node.
Rule 4: assigns a class to a leaf.

ID3 achieves rule 1 by applying an information entropy measure to choose the attribute that produces the partition with the least "randomness" in the distribution of classes (see [Quin86a] and [Chen88] for a detailed discussion). Rule 2 is realized by creating a subset for each value of the attribute chosen by rule 1. ID3 stops further partitioning of a node when all examples in it are all of one class, or when all the attributes have the same values for all the examples (rule 3). In case the examples of a leaf node are not all of the same class, assignment *by majority* or *by probability* may be used for rule 4[Brei84,Quin86b]. Rules 3 and 4 are trivially decided in the case of ID3 when the training set can be perfectly classified. Unfortunately, perfect classification of the training set is not always desirable. This is true, for instance, if the training data contains noise. Quinlan attempted tree pruning to deal with noise, and this has been somewhat successful [Quin86c]. But the problem is more serious when the input data is *inconclusive*. We discuss why pruning does not apply in this case in section 2.2.

2.1 A Simple Example

As an example of how overspecialization occurs on inconclusive data, consider the following expert system rules for diagnosing widget failures.

IF widget was dropped	IF widget was dropped
AND fell less than 5 ft.	AND fell more than 5 ft.
THEN replace casing.	THEN replace widget.

Let the two rules above be part of the *governing rule set* as defined in section 1. Now assume that ID3 is given the data set shown in figure 1, and that the attribute "height of widget's fall" is not provided to it. ID3 would then generate the tree shown in figure 1.

Obviously, the problem is that in order to obtain a tree which has only a single classification at each leaf node, it was necessary to specialize on irrelevant features. We would much rather see the specialization process stopped where the dotted line is drawn in the figure, thus producing the following probabilistic rule:

IF (Customer Abuse = dropped)
THEN Fix = replace casing (67%)
AND Fix = replace widget (33%).

This rule contributes useful knowledge about the proper classification. It is also an improvement over the ID3 output, which would ask the user for extraneous information and ultimately produce an unjustifiably narrow diagnosis. In section 3 we investigate how ID3 may be modified to reduce the overspecialization problem. Primarily, we are addressing the specification of rule 3 of the induction algorithm. We also propose more appropriate ways for realizing rules 1, 2, and 4.

2.2 Why Not Prune The Tree?

Inconclusive data sets are different from noisy data sets. In a noisy domain one can apply statistical techniques to prune away conditions (branches) which seem irrelevant to predicting a particular classification[Brei84, Quin86c,Quin87]. But for an inconclusive data set the

assumption upon which some pruning techniques are based, namely that there is a single correct classification for any combination of attribute values, is no longer valid. For pruning, some algorithms assume that a single class is the proper outcome of each rule and prune with respect to that outcome and thus remove preconditions of rules based on their statistical relevance to the single (usually most probable) outcome[Quin86c,Quin87]. This is clearly inappropriate if the attributes used can never allow perfect classification in the first place. In this case, it is not even correct for the algorithm to make the assumption that there should be only one class at each leaf node.

3 The INFERULE Algorithm

The INFERULE algorithm improves upon ID3 to make it more applicable to *inconclusive* data sets. This is achieved by adding a mechanism for deciding when to stop partitioning a given node in the tree (rule 3). For rule 4, INFERULE employs the probability method[Quin86b].

INFERULE differs from ID3 in that it generates strictly binary trees. This is a result of the fact that INFERULE specializes on the best attribute-value pair, rather than the best attribute, at any choice point. The advantage to this approach is that the data is never subdivided unnecessarily as is the case when ID3 specializes on an attribute having many values, only a few of which are actually relevant to diagnosing the failure. In [Chen88] a detailed discussion of weaknesses in ID3 resulting from this problem is provided.

3.1 Choosing An Attribute-Value Pair

We shall now turn our attention to the method used by INFERULE in selecting an attribute-value pair for partitioning the set of examples into two subsets. Let S be a set of examples. Each example consists of a specification of the values of all attributes and a class (outcome). The class is one of the m possible classes $\{c_1, \ldots, c_m\}$. Let A be an attribute that takes on one of the values $\{a_1, \ldots, a_k\}$. We now explain how INFERULE chooses one of A's values to partition S.

Let n_j be the number of examples in S that belong to class c_j. $S(a_i) \subseteq S$ is the subset of examples in S with value a_i for attribute A. $S(a_i, c_j) \subseteq S(a_i)$ is the subset of examples with $A = a_i$ and class c_j.

- $E_i(c_j) = \frac{|S(a_i)|}{|S|} n_j$ where $|S|$ denotes the number of examples in the set S. $E_i(c_j)$ is the *expected* number of examples in $S(a_i)$ that have class c_j. So $E_i(c_j)$ is an estimate of the actual value: $|S(a_i, c_j)|$.
- $SE(a_i) = \sqrt{\sum_{j=1}^{m} \frac{E_i(c_j)(n_j - E_i(c_j))}{n_j}}$. $SE(a_i)$ is the standard error [Quin86c] associated with the E_i's (estimates) defined above[2]. SE adjusts for the fact that an estimate that is based on a smaller data set is less accurate than one based on a larger set.
- $DI(a_i) = \sqrt{\sum_{j=1}^{m} [E_i(c_j) - |S(a_i, c_j)|]^2}$. DI is the geometric distance between the two outcome vectors: $\langle E_i(c_1), E_i(c_2), \ldots, E_i(c_m) \rangle$ and $\langle |S(a_i, c_1)|, |S(a_i, c_2)|, \ldots, |S(a_i, c_m)| \rangle$. Note that the first vector is the "expected" outcome vector of the subset $S(a_i)$, while the latter is the actual outcome vector of $S(a_i)$.

Finally, define

$$R(a_i) = SE(a_i) \cdot \frac{1}{DI(a_i)} \qquad (1)$$

R is the relative goodness of the attribute-value pair $A = a_i$. After evaluating all attributes and their values, the attribute-value pair with the minimum R value is the one chosen for partitioning the data. A property of R is that $R(a_i) = R(\neg a_i)$, i.e. the two subsets $S(a_i)$ and $S - S(a_i)$ evaluate equally in terms of their "distance" from S. Roughly stated, the goal of this criterion is to choose an attribute which maximizes the difference between the outcome vectors of the resulting subsets and the "expected" outcome vectors. In general, the greater this difference the more likely it is that the subset partition induced by $A = a_i$ is relevant to the classification. On the other hand, a small distance indicates that the distribution of classes in the original set does not significantly change when the set is partitioned. This would indicate that $A = a_i$ is probably not relevant to the classification task. The SE term allows for subsets of different sizes to be fairly compared. It adjusts for the sensitivity of the DI measure to small variations in $|S(a_i)|$ when the E_i's are small.

We illustrate the notion of geometric distance with the simple example of figure 2. The example shows two possible outcome vectors that may result when a set of 56 examples with 4 classes is to be partitioned using some attribute-value pair. It is assumed that half the examples in the set satisfy that attribute-value pair.

Note that this measure of attribute merit differs from ID3's information entropy measure. In the case of ID3, the entropy measures the discriminating power of an attribute by favouring attributes that result in outcome vectors that are unevenly distributed. Thus an attribute value is "bad" if its corresponding subset of examples has equal numbers of examples from different classes. In the

[2] As a matter of fact, the expression for the standard error SE as defined may easily be simplified to an expression which is a function of $|S(a_i)|$ only.

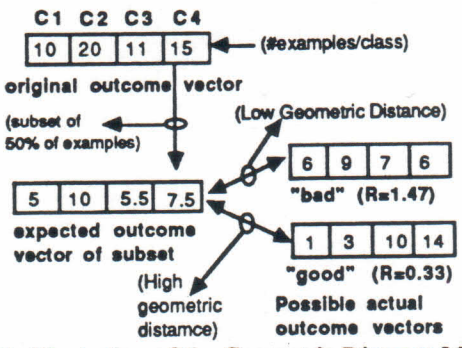

Figure 2: Illustration of the Geometric Distance Measure

case of INFERULE, the merit of an attribute-value pair is indicated by the fact that the class distribution in its corresponding subset differs significantly from the class distribution in the original set.

INFERULE's attribute selection criterion appeared to give the best results overall in comparison to several others which were also tried, including:

- Quinlan's information entropy measure [Quin86a]
- Chi-Squared test with $m - 1$ degrees of freedom[Brad76]
- Fisher's exact test for statistical independence using class compaction to reduce the input to a two class problem[Abra64]

3.2 To Specialize or Not To Specialize? That is The Question

INFERULE's "relative" measure of improvement in classification gives a natural way for deciding when to refrain from subdividing a given data set any further. By comparing the value of the measure R against a threshold T, one may control whether INFERULE should attempt to achieve a perfect classification or stop specialization and return a probabilistic guess of possible classes. INFERULE will halt specialization at a given node whenever the value of R exceeds the threshold T for every possible attribute-value pair.

Performance of the algorithm did not vary substantially with very small changes in the value of T, and overall it appeared that a value of $T = 0.75$ was nearly optimum for most data sets that we used. A formula for how T should be varied with respect to the number of attributes and attribute-values in a data set has not yet been determined. A setting of $T = 0.0$ would result in the generation of a single node tree which always predicts the most frequent class as an outcome. Setting $T = \infty$ results in the generation of a binary tree that classifies the examples in the training set perfectly. With this setting, any avoidance of overspecialization is disabled. With a setting of $T = 0.75$, INFERULE would indeed refrain from specializing the nodes below the dotted line in the tree of example given in figure 1.

4 Results

The data on which the INFERULE algorithm was tested contained records of automobile repair cases and how they were fixed. The data is made up of 8 symptom (attribute) fields with an average of 34 different values each. These symptom fields contain information on the make of car and engine as well as the year of the car and its mileage. They also contain codes for specific known problematic symptoms. This information is not sufficient to accurately diagnose the problem in most cases. Still, many useful conclusions can be drawn from this data.

Three data sets were constructed each containing different types of automotive repairs. The characteristics of each of these data sets are given below:

Data Set	Number of Cases	Avg. Size of Training Set	Possible Outcomes
Check Engine	14,175	12,700	44
No Start	4555	3650	32
Rough Engine	11,531	10,300	47

These three data sets were each randomly partitioned into training and test sets three different times. The results were then averaged over the different runs. The *accuracy* of a generated tree is defined to be the percentage of test cases for which the highest probability outcome suggested by the tree is the same as the actual outcome (class). Figure 3a shows the average accuracy for INFERULE, ID3 and ID3 with pessimistic pruning[Quin86c]. The fourth bar represents the frequency of the most common class and is given as a baseline for comparison[3]. We used a binary version of ID3 that uses only the best value of an attribute rather than branching on each attribute value. The binary ID3 performed better than the traditional ID3 algorithm, thus we do not show the results of the latter. Figure 3b shows the average number of rules obtained for each data set, and figure 3c shows the average runtimes of the three programs.

Two things the reader should note about these results. The first is that one of the primary advantages of the INFERULE approach, namely the ability to suggest several potential classifications, is ignored by the *accuracy* metric in Figure 3a, since it only considers whether or not the outcome suggested with the highest confidence agrees with the actual outcome. The percentage of test cases for which the actual outcome was among any of the ones suggested by INFERULE averaged between 80%

[3]This corresponds to running INFERULE with $T = 0.0$ as discussed in section 3.2

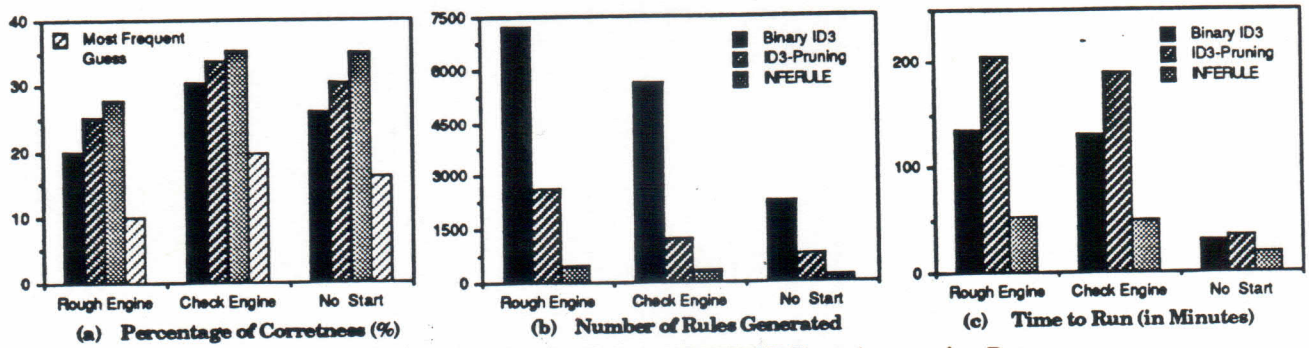

Figure 3: Test Results for ID3 and INFERULE on Automotive Data

and 90%. For ID3 with pruning the corresponding percentage was only 40-60%. So including multiple guessing would enhance INFERULE's performance to significantly higher levels than those shown in the graph. The second thing to note is that generating a full decision tree and then pruning it back to a smaller tree is a time consuming process. INFERULE generates an already pruned tree without incurring the extra cost of pruning.

5 Conclusions

The primary purpose of this paper is not to introduce yet another variant of ID3. Instead, we hope to influence future rule induction research to focus more on inconclusive data sets as opposed to conclusive data or data sets containing small amounts of random noise. After studying several diagnostic databases at General Motors, we are convinced that inconclusiveness is a characteristic that rule induction algorithms must be able to deal with if they are to have any practical application in industry.

The proper way of handling inconclusive data sets is to produce probabilistic, rather than categorical, classification rules. The assumption that rules with a single outcome are desirable should be abandoned if it is not statistically supported. This introduces the need for a criterion to determine when specialization is to be stopped before reaching a single classification. INFERULE addresses this issue and illustrates that the predicted improvement in performance is indeed achieved.

The most important area for future research is to find a good, universal criterion for halting the specialization process. The geometric distance test used in INFERULE has some of the properties of a good criterion, but may still lack generality with respect to different types of (non-automotive) data sets, especially those with a large number of attributes. Perhaps this paper will spark some interest in other research groups to try to discover this elusive criterion and thus produce a truly practical rule induction algorithm.

Acknowledgements

The authors would like to acknowledge the help of Andrew Chou in the implementation of INFERULE and in formulating the distance measure used. Usama M. Fayyad is supported by a Rackham Predoctoral Fellowship and by the EECS Department of The University of Michigan.

References

[Abra64] M. Abramowitz and I. Stegun. *The Handbook of Mathematical Functions with Formulas, Graphs and Mathematical Tables*. National Bureau of Standards. (1964).

[Brad76] J.V. Bradely. *Probability; Decision; Statistics*. pp. 278-281. Prentice Hall Inc. Englewood Cliffs, NJ (1976).

[Brei84] L. Breiman, J.H. Friedman, R.A. Olson, and C.J. Stone. *Classification and Regression Trees*. Wadsworth & Brooks (1984).

[Chen88] J. Cheng, U.M. Fayyad, K.B. Irani, and Z. Qian. "Improved Decision Trees: A Generalized Version of ID3." *Proceedings of the Fifth International Conference on Machine Learning*. pp. 100-107. Morgan Kaufmann, (1988).

[Quin86a] J. R. Quinlan. "Induction of Decision Trees." *Machine Learning 1. No. 1*. pp. 81-106. 1986.

[Quin86b] J.R. Quinlan. "The effect of noise on concept learning." In *Machine Learning: An Artificial Intelligence Approach, vol. 2*, Michalski *et al* (Eds). Tioga Publishing Company, (1986).

[Quin86c] J. R. Quinlan. "Simplifying Decision Trees." MIT AI Memo No. 930, (1986).

[Quin87] J. R. Quinlan. "Generating Production Rules from Decision Trees." *IJCAI-87*. pp. 304-307. Milan, Italy. (1987).