

An Oracle based Meta-Learner for ID3

R. Syama Sundar Yadav and Deepak Khemani

A.I.D.B. Lab,

Dept. of Computer Science and Engineering,

Indian Institute of Technology Madras,

Chennai 600036, India.

shyam@cs.iitm.ernet.in, khemani@iitm.ac.in

Abstract

The quality of a learning algorithm is characterized by the accuracy, stability and comprehensibility of the models it generates. Though ensembles produce accurate and stable classifiers, they are hard to interpret. In this paper, we propose a meta-learning method for ID3 that makes use of an ensemble for gaining accuracy and stability and yet produces a single comprehensible classifier. The main idea here is to generate additional examples at every stage of the decision tree construction process and use them to find the best attribute test. These new examples are classified using the ensemble constructed from the original training set. The number of new examples generated depends on the size of the input attribute space and the input attribute values of new examples are partially determined by the algorithm. Existing work in this area deals with the generation of a fixed number of random examples. Experimental analysis shows that our approach is superior to the existing work in retaining accuracy and stability gains provided by the ensemble classifier.

Introduction

Machine learning deals with the design of algorithms that automatically extract useful knowledge from past experiences. Mathematically, given a set of training examples that partially describes a function $y = f(X)$, the learning algorithm's task is to output a classifier that approximates the true function f and predict the y value for an unseen X . Several learning algorithms have been proposed in the literature that largely vary by the way they represent the final classifier. Examples are ID3 (Quinlan 1993a), CN2 (Clark & Niblett 1989), Neural Networks (Gallant 1993). Though these algorithms are experimentally proved to produce accurate classifiers on a collection of real world examples, they are over responsive to training data; i.e., with small changes in the training data, they produce entirely different classifiers.

Learning multiple models (also called ensembles) for reducing instability as a means to improve accuracy of learning algorithms has been an active line of research (Dietterich 1997). The idea here is to learn several different

models by varying the learner or the training data and then combine these models in some way (voting) to make predictions. Different forms of this approach include bagging (Breiman 1996), boosting (Freund & Schapire 1996) and stacking (Wolpert 1992). Though this approach improves stability and accuracy, it gives up the essential characteristic of a learner, namely output comprehensibility. Understanding the several models produced by this approach and keeping track of how they interact to predict a new case is a hard task for the user. There has been substantial work on improving the comprehensibility of learned classifiers (Quinlan 1993b) and (Craven & Shavlik 1996) since users often wish to gain insight into a domain rather than simply obtain an accurate classifier for it. This is possible only if they are able to understand the learner's output. Even when predictive accuracy is the sole goal, comprehensibility is an important asset for a learner, because it facilitates the process of interactive refinement that is crucial for most practical applications (Domingos 1997).

In this paper, we present an oracle based meta-learning method called *oracleID3* for ID3 (Quinlan 1993a) that generates additional examples at every stage of the decision tree construction process, classifies them using a bagged ensemble, adds them to the training data and then induces a single comprehensible decision tree. This method is inspired from Combined Multiple Models (CMM, for short) approach proposed by (Domingos 1997). The main idea behind this approach can be summarized as follows: In general, when the training data is sparse, the learning algorithm's heuristics (information gain in the case of ID3) may not allow it to find the accurate classifier. The learning algorithm's heuristics may benefit if the algorithm is given some additional examples. The class values of these additional examples can be found from an ensemble constructed from the same training data, as the ensemble is shown to be more accurate than single classifier (Breiman 1996). It is also shown that accuracy and stability of learned models tend to increase with the training set size (due to decreasing variance) (Kohavi & Wolpert 1996).

However, the proposed method differs from CMM in two principal ways. In CMM, the values of new examples are generated randomly following the distribution inherent in the classifier produced by the bagged ensemble and the number of new examples is set to 1000 irrespective of the size of

the training set. In the proposed method, the new examples are added at every stage of the decision tree construction process and the values of input attributes of new examples are partially discovered by the algorithm. The number of new examples generated depend on the domain size of training data and vary from one dataset to other. These two modifications are necessary in order to properly guide the heuristics of the learner and to vary the number of new examples according to training set's domain size, as generating too many examples may mislead the meta-learning algorithm in the case of small attribute spaces. It is also pointed out in (Domingos 1997) that the number of new examples need to be derived from the dataset size in order to produce less complex models. The proposed method is empirically evaluated to verify its superiority to CMM approach in retaining accuracy and stability gains provided by bagged ensemble.

The rest of the paper is organized as follows: Section 2 reviews the related work while Section 3 presents our meta-learning method. Section 4 evaluates the method proposed and Section 5 finally concludes by shedding light on future directions.

Related Work

Making the classifier produced by a learner simpler and more comprehensible has been the prominent direction of research in inductive learning. Work by (Quinlan 1993b) concerns with the production of simpler decision trees apart from its effect on accuracy. There has been some focus on extracting single, comprehensible decision tree from multiple decision trees. (Quinlan 1987) describes merging all branches from multiple decision trees into a single rule set and extracting the best rules. (Buntine 1990) describes a method to extract a single good decision tree from an option tree (Kohavi & Kunz 1997). (Shannon & Banks 1997) proposed a method for combining multiple decision trees into one, based on measuring distances between them and finding the median tree.

A meta-learning approach (called as CMM) for extracting comprehensible decision trees from ensembles was proposed by (Domingos 1997). Here, a bagged ensemble is constructed from the original training set and some fixed number of new examples are generated and added to the original training set. The output values of new examples are found using the ensemble. The main idea here is that increasing the size of the training set decreases the variance thus resulting in more stable classifiers. As this meta-learner learns a single decision tree from the original and new dataset, the classifier it produces is comprehensible when compared to that of bagged ensemble.

CMM is an example of an approach for extracting comprehensible output from a learned model. Substantial research has been carried out in the case of neural network (Towell & Shavlik 1993) and (Andrews & Dietterich 1996). Algorithms based on queries to an oracle are also relevant to this problem. For example, work by (Craven & Shavlik 1996) uses an already learned neural network model as an oracle and learns a comprehensible decision tree. The main focus of this work is on generating symbolic knowl-

edge from neural networks without losing much on accuracy.

Proposed Meta-Learning Algorithm

In this section, we first briefly describe the ID3 algorithm, then analyze the reasons for instability of the models it produces and finally present our meta-learning method.

ID3

Given a set of training examples of the form $(x_1, x_2, \dots, x_m, y)$ that partially describes some unknown function, $y = f(X)$, ID3 produces a classifier that approximates the true function, f , in the form of a decision tree. Each internal node of a decision tree describes a test on one of the input attributes, x_i , and an edge emanating from a node represents an outcome of the test at that node. Leaves specify the output class of examples associated with it. The value of a test case is found by propagating it down the decision tree following the path satisfied by the input attributes of the test case till it reaches the leaf; the value of the test case is then predicted as that of the leaf.

The crux of ID3 algorithm lies in finding the attribute tests at each internal node of the tree. Every node of the tree is associated with a subset of training examples and input attributes. The example set present in each node is partitioned according to a test on each input attribute associated with that node. Each test is then evaluated based on a measure called information gain. The test that gives the maximum information gain is chosen for the current node and children nodes are created depending the number of outcomes of the test. Each child is populated with those examples present in the current node which satisfy the outcome of the test. The method is repeated till a node contains all examples that have same output value or no further test can be found that results in information gain.

As (Ali 1996) points out, the main reason for instability of ID3 is that a candidate with the highest information gain is flanked by other candidates that have "almost as much" information gain. The candidate that is truly the best appears to be second best due to the inclusion or exclusion of a few examples. Moreover, a small change in one split close to the root will change the whole subtree below (Breiman 1994). So, a small change in the training data can drastically change the decision tree learned.

(Kohavi & Kunz 1997) propose a method that mitigates the above said problem by including option nodes in the decision tree. An option node is like an *or* node in *and-or* trees. It contains tests on more than one attribute and represents uncertainty in the decision process. Though it is shown to increase the stability of ID3, option trees are hard to interpret when compared to a single decision tree, as the user has to keep track of several subtrees when predicting the value of a new case.

Oracle ID3

In this sub section, we present our oracle based meta-learner for ID3 that tries to avoid above mentioned drawbacks by generating additional examples that would help discover the

correct attribute test at each node of the decision tree. The proposed meta-learning method, shown in the Algorithm 1, mainly consists of three modules: *Oracle*, *Tree Expander* and *Example Generator*. Example generator finds the input values of new examples while the class values of these additional examples are predicted with the help of an ensemble (called Oracle). The main purpose of Tree expander is to evaluate all the attribute tests and choose the best one. The details of three modules are given in the following.

Algorithm 1 oracleID3(*training_examples*)

```

Queue ← ∅
initialize the root of the tree, T, as a leaf node
put(T, training_examples) into Queue
construct bagged ensemble (Oracle) from
training_examples
while Queue is not empty do
  remove (N, examplesN) from head of Queue
  find k candidate attribute tests that result in maximum
  information gain
  for each candidate test, t do
    let xt be the attribute tested in t
    partition examplesN according to outcomes of t
    for each outcome, o, of the test, t do
      construct n (equal to size of the partition) new ex-
      amples in the following way:
      let p be the path from root of the tree to the current
      node, N
      let X be the attributes tested at the nodes on the
      path p
      let V be the labels of the edges on the path p
      set the values of the attributes X to V in each new
      example
      set the value of xt to the outcome, o
      set the values of remaining attributes to one of the
      possible respective domain values randomly
      use Oracle to find the output value of each new
      example
      add the new examples thus constructed in the cur-
      rent partition
    end for
  end for
  re-evaluate candidate tests
  select the best test, tbest
  for each outcome, o, of tbest do
    make C, a new child node of N
    examplesC ← members of examplesN with out-
    come o on test, tbest
    put (C, examplesC) into Queue
  end for
end while
return T

```

Oracle The main role of oracle is to predict the class values of new examples. Here the oracle is a bagged ensemble constructed from the original training set using ID3 as the base learning algorithm. The number of models (decision

trees) generated in bagging is set to 10. The new examples presented to the oracle are complete in the sense that the values are given for all input attributes.

Tree Expander Given a node of the tree and a set of training examples associated with it, the tree expander’s task is to expand the node by choosing the correct attribute test for that node. Finding the attribute test for a given node is a two step process. First, tree expander finds the top *k* attribute tests that result in the partitions having maximum information gain. It then passes these partitions and the partially constructed tree to the example generator which adds additional examples in all the partitions. The second step is to re-evaluate each candidate attribute test and pick the best partition.

Example Generator The main task of this module is to find the input attribute values of new examples. Input to this module consists of the partial tree (possibly empty) so far constructed and the candidate attribute tests at the current node. For each candidate attribute test, *t*, it adds *k_p* new examples to each partition, *p* (resulted due to the attribute test, *t*) where *k_p* is the size of the partition, *p*. The values of new examples are given by the labels on the path from the root to the current node. Note that, this path does not specify values for all input attributes; the remaining attributes are generated randomly. The output value of each new example is found from the oracle.

The main idea here is to re-evaluate the top partitions at each node by providing additional information in the form of new examples and pick the correct attribute test. Here the number of candidate tests examined is set to 3 in accordance with the empirical results provided in (Kohavi & Kunz 1997). The number of examples generated at each node is found by some preliminary experimental analysis and is set to the size of the partition in which new examples are added.

Empirical Evaluation

This section presents empirical evaluation of our method. The question of whether the proposed method is superior to CMM approach in retaining accuracy and stability gains provided by ensemble classifier has to be answered experimentally. The underlying ID3 algorithm we implemented was the basic method proposed in (Quinlan 1993a), which does not deal with the missing values. Ensemble was constructed using bagging technique (Breiman 1996) with ID3 as the base learner. In our implementation of CMM approach to ID3, the number of new examples was set to 1000.

The experiments are carried out on 9 datasets taken from UCI Machine learning repository (Merz, Murphy, & Aha 1997). The characteristics of the datasets chosen for experimental analysis are shown in Table 1. There are basically two kinds of datasets: full training sets taken from large domains and a small portion (10%) of available training data taken from small domains. The datasets Breast-Cancer, Lung-Cancer, Cancer and Zoo fall in the first category while Monk1, Monk2, Monk3, Car and Nursery come

Dataset	#examples	#attrs	#classes	#val/attrs
Cancer	683	9	2	10
Breast-Cancer	277	9	4	5.67
Lung-Cancer	33	56	2	4
Zoo	101	16	1	2
Monk1	24	6	2	2.8
Monk2	84	6	2	2.8
Monk3	122	6	2	2.8
Car	172	6	4	3.5
Nursery	1296	8	5	3.4

Table 1: Characteristics of the datasets used in experimental study

Dataset	ID3	bagged ID3	oracle ID3	CMM ID3
Breast-Cancer	58.84 \pm 2.3	67.65 \pm 4.19	65.52 \pm 2.77	60.64 \pm 2.32
Lung-Cancer	78.89 \pm 3.56	85.19 \pm 5.63	83.33 \pm 4.1	81.85 \pm 3.48
Cancer	91.1 \pm 4.53	92.68 \pm 7.32	92.82 \pm 5.52	92.15 \pm 6.3
Zoo	90.72 \pm 5.6	94.1 \pm 9.8	93.91 \pm 8.22	92.97 \pm 8.4
Monk1	78.37 \pm 7.1	81.6 \pm 13.7	80.23 \pm 10.83	76.74 \pm 9.63
Monk2	58.14 \pm 8.2	65.11 \pm 10	63.25 \pm 10.32	61.86 \pm 9.53
Monk3	90.7 \pm 12.25	95.12 \pm 10	95.34 \pm 9.80	94.41 \pm 9.60
Car	77.67 \pm 4.7	82.56 \pm 8.74	81.16 \pm 10.12	80.23 \pm 9.74
Nursery	88.97 \pm 1.7	90.74 \pm 3.37	89.33 \pm 3.43	88 \pm 3.68

Table 2: Accuracy results of ID3, baggedID3, oracleID3 and CMMID3

under second category. The second kind of datasets (small domains) are chosen in order to verify whether keeping the number of new examples dependent on the size of the attribute space would result in more accurate classifiers. The reason for taking only a portion of training data in the case of small domains is to assess the usefulness of additional examples when learning from sparse training data.

Classification accuracies are measured using 10-fold cross validation. The average and standard deviation of accuracies for all methods are shown in Table 2. From the table, it is evident that accuracy gains retained by our approach are more than that of CMM, for the datasets chosen. On the average, our approach loses only 1.2% of accuracy gains provided by the bagged ensemble while CMM loses 3.1%. OracleID3 is more accurate than ID3 and CMMID3 with a confidence of 90% according to a paired *t*-test. Table 3 shows the number of new examples generated by our approach for each dataset. It is clear from the table that new examples are generated according to the size of the training set. Note that at every level, additional examples are added for several candidate tests. The number indicated here is the number of new examples finally added to the original training data, meaning the sum of the number of examples added for the best attribute test found after re-evaluation, at all levels.

Note that in the case of large domains, the upper bound didn't allow our algorithm to generate more than 1000 new examples. In the case of small domains, our approach clearly dominated CMM even though it generated less number of examples compared to CMM. This is a clear evidence of our claim that producing suitable new examples, instead

of random examples, would promote the meta-learning algorithm to induce more accurate classifiers.

The stability gains provided by oracleID3 and CMMID3 are shown in Table 4. The average stability gains lost by oracleID3 is 3.73% while CMMID3 loses 5.34% of the gains provided by bagged ensemble. Moreover, oracleID3 produced more stable classifiers than CMMID3 in all the datasets. Table 5 shows the sizes of the decision trees for all the methods compared. In case of baggedID3, the size reported is sum of the sizes of all decision trees (10, in this case) learned by bagging. The table shows that sizes of the trees learned by oracleID3 and CMMID3 are comparable to those of normal decision trees while the trees learned by baggedID3 are more than 10 times larger. Moreover, the trees learned by oracleID3 are smaller than those induced by CMMID3 in all but one case in which the sizes are almost equal. From these results, it can be implied that our method is superior to CMM approach for inducing accurate, stable and comprehensible classifiers.

Conclusions

In this paper we presented a novel meta-learning approach for ID3 to induce comprehensible classifiers from ensembles. The existing work (Domingos 1997) in this area deals with the generation of some fixed number of random examples. The main contribution of the proposed method is the automatic discovery of the values of new examples to be generated and variation of the number of new examples according to the size of the training set. The advantages of these two modifications to the existing work are experimentally verified.

Dataset	no.of New Examples
Breast-Cancer	1000
Lung-Cancer	1000
Cancer	1000
Zoo	1000
Monk1	332
Monk2	617
Monk3	202
Car	709
Nursery	1000

Table 3: Number of Additional Examples Generated by oracleID3

Dataset	ID3	bagged ID3	oracle ID3	CMM ID3
Breast-Cancer	78.1	84.77	82.03	81.93
Lung-Cancer	62.9	75.02	70.2	67.11
Cancer	59.3	75.75	64.71	63.43
Zoo	60.37	73.37	70.74	66.82
Monk1	74.72	80.9	79.22	79.13
Monk2	70.93	78.04	75.81	75.74
Monk3	74.53	89.56	83.97	79.33
Car	70.34	81.46	81.28	79.6
Nursery	78.92	86.89	86.5	86.01

Table 4: Stability results of ID3, baggedID3, oracleID3 and CMMID3

The proposed method can be seen as extracting comprehensible classifiers from black-box models (that are proved to be more accurate) without losing much on accuracy. So, our approach can be directly extended from ensembles to any other accurate and incomprehensible classifiers like Neural Networks (Gallant 1993). There were some methods proposed in the literature for extracting symbolic rules (Craven & Shavlik 1993) and decision trees (Craven & Shavlik 1996) from learned neural networks as the latter proved to be more accurate than symbolic classifiers. We are now focussed on extending our approach to extract decision trees from neural networks and testing how well it compares to the existing methods.

References

- Ali, K. 1996. *Learning probabilistic Relational Concept Descriptions*. Ph.D. Dissertation, University of California, Irvine.
- Andrews, R., and Dietterich, J., eds. 1996. *Proc. NIPS-96 Workshop on Rule Extraction from Trained Artificial Neural Networks*. Snowmass, CO: The NIPS Foundation.
- Breiman, L. 1994. Heuristics of instability in model selection. Technical report, University of California at Berkeley.
- Breiman. 1996. Bagging predictors. *Machine Learning* 24(2):123–140.
- Buntine, W. 1990. *A Theory of Learning Classification Rules*. Ph.D. Dissertation, School of Computing Science, University of Technology, Sydney, Australia.
- Clark, P., and Niblett, T. 1989. The cn2 induction algorithm. *Machine Learning* 3:261–283.
- Craven, M., and Shavlik, J. 1993. Learning symbolic rules using artificial neural networks. In *Proc. of the 10th International Conference on Machine Learning*, 73–80. Amherst, MA: Morgan Kaufmann.
- Craven, M., and Shavlik, J. 1996. Extracting tree-structured representations of trained networks. *Advances in Neural Information and Processing Systems* 8:24–30.
- Dietterich, T. 1997. Machine learning: Four current directions. *AI Magazine* 97–136.
- Domingos, P. 1997. Knowledge acquisition from examples via multiple models. In *Proc. of the Fourteenth Intl. Conf. on Machine Learning*, 98–106. Morgan Kaufmann.
- Freund, Y., and Schapire, R. 1996. Experiments with a new boosting algorithm. In *Proc. of the Thirteenth Intl. Conf. on Machine Learning*, 148–156. Morgan Kaufman.
- Gallant, S. 1993. *Neural Network Learning and Expert Systems*. MIT Press.
- Kohavi, R., and Kunz, C. 1997. Option decision trees with majority votes. In *Fourteenth International Conference on Machine Learning*. Morgan Kaufmann.
- Kohavi, R., and Wolpert, D. 1996. Bias plus variance decomposition for zero-one loss functions. In *Thirteenth International conference on Machine Learning*, 275–283. Bari, Italy: Morgan Kaufmann.
- Merz, C.; Murphy, P.; and Aha, D. 1997. *UCI Repository of Machine Learning Resources*. Department of Information and Computer Science, University of California at Irvine.
- Quinlan, J. 1987. Generating production rules from decision trees. In *Proc. Tenth International Joint Conference*

Dataset	ID3	bagged ID3	oracle ID3	CMM ID3
Breast-Cancer	438.7	5278.8	420.38	618.41
Lung-Cancer	12.2	192.8	112.8	111.52
Cancer	191	2538	280.9	468.3
Zoo	22.6	416.4	32.3	35.76
Monk1	24.7	373.8	37.31	54.82
Monk2	31.8	373.8	58.91	75.59
Monk3	16.9	280.8	19.96	24.54
Car	78.6	1110.6	142.75	174.02
Nursery	329.1	4781.4	343.68	409.71

Table 5: Comprehensibility results of ID3, baggedID3, oracleID3 and CMMID3

on *Artificial Intelligence*, 304–307. Milan, Italy: Morgan Kaufmann.

Quinlan, J. 1993a. *C4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann.

Quinlan, J. 1993b. *Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann: Morgan Kaufmann. chapter 5.

Shannon, W., and Banks, D. 1997. A distance metric for classification trees. In *Proc. Sixth International Workshop on Artificial Intelligence and Statistics*, 457–464. Fort Lauderdale, FL: Society for Artificial Intelligence and Statistics.

Towell, G., and Shavlik, J. 1993. Extracting refined rules from knowledge-based neural networks. *Machine Learning* 13:71–101.

Wolpert. 1992. Stacked generalization. *Neural Networks* 5:241–259.