Error Reduction through Learning Multiple Descriptions

KAMAL M. ALI AND MICHAEL J. PAZZANI ali@ics.uci.edu, pazzani@ics.uci.edu
Department of Information and Computer Science, University of California, Irvine, CA 92717.

Editor: Lorenza Saitta

Abstract. Learning multiple descriptions for each class in the data has been shown to reduce generalization error but the amount of error reduction varies greatly from domain to domain. This paper presents a novel empirical analysis that helps to understand this variation. Our hypothesis is that the amount of error reduction is linked to the "degree to which the descriptions for a class make errors in a correlated manner." We present a precise and novel definition for this notion and use twenty-nine data sets to show that the amount of observed error reduction is negatively correlated with the degree to which the descriptions make errors in a correlated manner. We empirically show that it is possible to learn descriptions that make less correlated errors in domains in which many ties in the search evaluation measure (e.g. information gain) are experienced during learning. The paper also presents results that help to understand when and why multiple descriptions are a help (irrelevant attributes) and when they are not as much help (large amounts of class noise).

Keywords: Multiple models, Combining classifiers

1. Introduction

Learning multiple models of the data has been shown to improve classification error rate as compared to the error rate obtained by learning a single model of the data (for example: decision trees: Kwok & Carter, 1990; Buntine, 1990, Kong & Dietterich, 1995; rules: Gams, 1989; Smyth & Goodman, 1992; Kononenko & Kovacic,1992; Brazdil & Torgo, 1990; neural nets: Hansen & Salamon, 1990; Baxt, 1992; Bayesian nets: Madigan & York, 1993; regression: Perrone, 1993, Breiman, in press). Although much work has been done in learning multiple models not many domains were used for such studies. There has also been little attempt to understand the variation in error reduction (the error rate of multiple models compared to error rate of the single model learned on the same data) from domain to domain. Three of the data sets used in our study for which this approach provides the greatest reduction in error (Tic-tac-toe, DNA, Wine) have not been used in previous studies. For these data sets, the multiple models approach is able to reduce classification error on a test set of examples by a factor of up to seven! This paper uses a precise definition of "correlated errors" to provide an understanding of the variation in error reduction. We also present the idea of "gain ties" to understand why the multiple models approach is effective - especially, why it is more effective for domains with more irrelevant attributes.

Machine Learning, VolNum, 101–132 (Year)

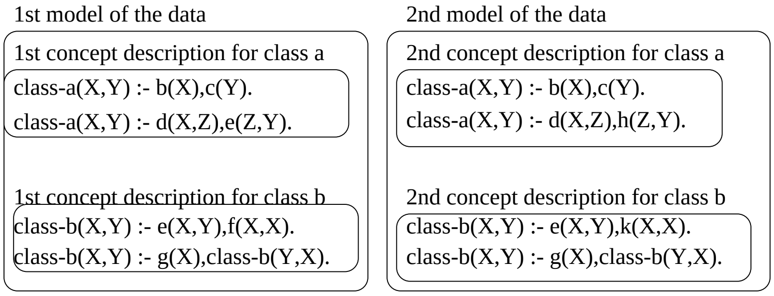


Figure 1. An example of learning multiple models - each model consists of a set of class descriptions.

Figure 1 shows an example of multiple learned models of the form used in this paper. In the multiple models approach, several models of one training set are learned. Each model consists of a description for each class. Each description is a set of rules for that class (i.e. each class description is a set of first-order Horn clauses[1] for that class). The set of learned models is called an ensemble (Hansen & Salamon, 1990).

Previous work in learning multiple models has mainly been concerned with demonstrating that the multiple models approach reduces error as opposed to the goal of this paper which is to explain the variation in error reduction from domain to domain. Previous work has compared different search strategies (Kononenko & Kovacic, 1992) compared different search evaluation measures (Gams, 1989; Smyth & Goodman, 1992), evaluated the effects of pruning (Kwok & Carter, 1990; Buntine, 1990) and compared different ways of generating models (nearly all authors). Except for the work of Buntine, all the other comparisons have been made on a few domains so we still do not have a clear picture of how domain characteristics affect the efficacy of using multiple models. It is important to analyze these experimental data because the amount of error reduction obtained by using multiple models varies a great deal. On the wine data set, for example, the error obtained by uniformly weighted voting between eleven, stochastically-generated descriptions is only one seventh that of the error obtained by using a single description. On the other hand, on the primary-tumor data set, the error obtained by the identical multiple models procedure is the same as that obtained by using a single description.

Much of the work on learning multiple models is motivated by Bayesian learning theory (e.g. Bernardo & Smith, 1994) which dictates that to maximize predictive accuracy, instead of making classifications based on a single learned model, one should ideally use all hypotheses (models) in the hypothesis space. The vote of each hypothesis should be weighted by the posterior probability of that hypothesis given the training data. Since the theory requires voting from all hypotheses or models in the hypothesis space, all tractable implementations of this theory have to be approximations. This raises the following experimental question: what model-generation/evidence-combination method yields the lowest error rates in practice? Or, how can one characterize the domains in which a particular method works best and why does it work best on such domains?

The main hypothesis examined in this paper is whether error is most reduced for domains for which the errors made by models in the ensemble are made in an uncorrelated manner. In order to test this hypothesis, we first need to define error reduction more precisely. Two obvious measures comparing the error of the ensemble (E_e) to the error of the single model (E_s) are error difference ($E_s - E_e$) and error ratio ($E_r = E_e/E_s$). We use error ratio because it reflects the fact that it becomes increasingly difficult to obtain reductions in error as the error of the single model approaches zero. Error ratios less than 1 indicate that multiple models approach was able to obtain a lower error rate than the single model approach. The lower the error ratio, the greater the error reduction. A precise definition of the notion of "correlated errors" is presented in Section 5.2. Briefly, our metric (ϕ_e, "fraction of same (correlated) errors") measures the proportion of the test examples on which members of an ensemble make the same kinds of misclassification errors. Two models are said to make a "correlated error" when they both classify an example of class i as belonging to class $j, j \neq i$.

The paper presents results on why it is possible to learn models with more uncorrelated errors for some domains than for others. We also explore the effect of varying two domain characteristics (level of class noise and number of irrelevant attributes) on error ratio. Finally, we examine the effect of syntactic diversity on ensemble error. This follows the work of Kwok & Carter (1990) which postulates that learning more syntactically diverse decision trees leads to lower ensemble error.

The remainder of the paper is organized as follows. After an examination of the main issues in learning multiple models, we present our core learning algorithm HYDRA (Ali & Pazzani 1992,1993,1994) which we modify in various ways to learn multiple models. Next, we present results of experiments designed to answer the following questions:

1. What effect does the multiple models approach have on classification error as compared to the error produced by the single model learned from the same training data?

2. What is the relationship between the amount of observed error reduction (E_r) and the tendency of the learned models to make correlated errors?

3. Can the amount of error reduction observed for a domain be *predicted* from the number of ties in gain experienced by the learning algorithm on that domain?

4. How does increasing the amount of class noise affect the amount of error reduction?

5. How does increasing the number of irrelevant attributes affect the amount of error reduction?

6. Does increasing the diversity of the models *necessarily* lead to greater reduction in error?

2. Background

Previous empirical work in using multiple models (e.g. Buntine, 1990; Kononenko & Kovacic, 1992) has mainly focused on demonstrating error reduction through using multiple models and exploration of novel methods of generating models and combining their classifications. The work can be characterized along three dimensions: the kind of model being learned (tree, rule etc.), the method of generating multiple models, and the method of combining classifications of the models to produce an overall classification. The work of Kwok & Carter (1990) also serves as foundation for our work on the effect of syntactic diversity on error rate. They showed that ensembles with decision trees that were more syntactically diverse obtained better accuracies than ensembles with trees that were less diverse.

Previous theoretical work in learning multiple models includes Buntine's formulation of general Bayesian learning theory, Schapire's (1990) Boosting algorithm and the results from Hansen & Salamon (1990) and Drobnic & Gams (1992, 1993). Schapire's work proceeds on the basis (proved in Hansen & Salamon, 1990) that models that make errors in a completely independent manner will produce lower ensemble error. His Boosting algorithm is the only learning algorithm which incorporates the goal of minimizing correlated errors during learning. However, the number of training examples needed by that algorithm increases as a function of the accuracy of the learned models. Schapire's method could not be used to learn many models on the modest training set sizes used in this paper.

Other theoretical results on the effects of using multiple models come from Hansen & Salamon (1990) who prove that if all models have the same probability of making an error, and this probability is less than 0.5 and if they all make errors completely independently then the overall error must decrease monotonically as a function of the number of models. Theoretical analysis of using multiple regression models has also been done by Breiman (in press). However, this research does not say anything about the amount of error reduction and Hansen and Salamon's research does not say anything when errors are not completely independent.

With the exception of Buntine (1990), most of the empirical work has been done on a small number of domains (two: Kwok & Carter (1990); three: Kononenko & Kovacic (1992); three: Smyth *et al.* (1990)). The small number of domains used reduces the chance of accurately characterizing the conditions under which the method works. Furthermore, although Buntine used many data sets, he did not try to explain the variation in error reduction. By using twenty-nine data sets from twenty-one domains we are better able to study what domain characteristics are factors in error reduction (a data set being different from a domain in that it also involves specifying parameters such as number of training examples, noise levels and irrelevant attributes).

```
FOIL(POS-EGS,NEG-EGS,Positive-class-name,Arity):
    Let LearnedDescription be the empty set
    Until POS-EGS is empty do
          Separate: (begin a new clause)
          Let head of NewClause be Positive-class-name(V_1,...,V_Arity)
          Let body of NewClause be empty, NEG be NEG-EGS, POS be POS-EGS
          Until NEG is empty do:
                Conquer: (build a clause body)
                Conjoin to body of NewClause the literal that yields highest gain
                Remove from POS and NEG examples that do not satisfy NewClause
          End
          Add NewClause to LearnedDescription
          Remove from POS-EGS all positive examples that satisfy NewClause.
    Return LearnedDescription
```

Table 1. Pseudo-code for FOIL.

3. Methods for learning multiple class descriptions

We consider two methods for generating multiple class descriptions: stochastic hill-climbing (Ripley, 1987; Kononenko & Kovacic, 1992) and deterministic learning from a k-fold partition of the training data (Gams, 1990). Although these methods are not new, our goal is to show that our results pertaining to error reduction, correlatedness of errors and gain ties apply to more than one method of generating multiple models.

We use HYDRA (Ali & Pazzani, 1993) to learn a single model consisting of a description for each class. HYDRA is based on extensions to FOIL[2] (Quinlan, 1990) proposed in Ali & Pazzani (1993) and Pazzani *et al.* (1991). HYDRA is then further modified to learn several models.

The pseudo-code for FOIL is presented in Table 1. FOIL learns one clause (rule) at a time, removing positive training examples covered by that clause in order to learn subsequent clauses. This is referred to as the "separate and conquer" (Quinlan, 1990) or "covering" (Michalski & Stepp, 1983) strategy. The basic FOIL procedure learns as follows. A clause for a given class such as *class-a* is learned by a greedy search strategy. It starts with an empty clause body which covers all remaining positive and negative examples. Next, the strategy considers all literals that it can add to the clause body and ranks each by the information gained (Quinlan, 1990) if that literal were to be added to the current clause body. Briefly, the information gain measure favors the literal whose addition to the clause body would result in a clause that would cover many positive examples and exclude many negative examples. The literal that yields the highest information gain is then added to the clause body. The strategy keeps adding literals until either the clause covers no negative examples or there is no candidate literal with positive information gain. Positive examples covered by the clause are removed from the training set and the process continues to learn subsequent clauses on the remaining examples, terminating when no more positive examples are left.

FOIL only learns in data sets consisting of two-classes, one of which must be identified as the "positive" class. FOIL learns a class description only for the class identified as the "positive" class. Thus, FOIL learns a single model consisting of a single class description. FOIL uses the closed-world assumption (Lloyd, 1984) for classification: if the test example matches the body of any clause learned for class "positive" then the example is assigned to class "positive." If it fails to match *any* clause, FOIL uses the closed-world assumption and assigns the example to class "negative."

The way we extend FOIL to learn a rule set for each class is by treating examples of all other classes as negative. This is the algorithm used in HYDRA (Ali & Pazzani, 1993). We prefer this way of learning for multi-class data rather than learning a set of rules of the form:

$$\begin{aligned}\text{class}(V_1 ... V_n,\ \text{X}) &\leftarrow \ ...,\ \text{X} = \text{class-a}\\ \text{class}(V_1 ... V_n,\ \text{X}) &\leftarrow \ ...,\ \text{X} = \text{class-b}\end{aligned}$$

because of a technical limitation with FOIL - there is no guarantee in FOIL that the variable corresponding to the class (X) will appear in the body of the learned clause. Now we discuss two methods of learning several descriptions for each class in the training data. These methods involve executing the HYDRA procedure once for each model to be learned.

- **Stochastic Hill-climbing-** Stochastic hill-climbing only involves modifying HYDRA's procedure for selecting which literal to add to the clause currently being learned. Instead of picking the best literal (ranked according to some measure such as information gain) stochastic hill-climbing stores all literals that are within some margin, β, of the best and then picks non-deterministically from among that set. The probability of a literal being picked is proportional to its gain. The set of literals whose gain exceeds β times that of the best literal is called the "bucket."

- **k-fold partition learning-** This procedure generates k models by partitioning the training data into k equal-sized sets and in turn, training on all but the i-th set. HYDRA is called k times and each time it learns a class description for each class in the data set. k-fold partition learning was first used by Gams (1989) whose system learns ten models using 10-fold partition learning and then combines them into a single model. By doing so, however, he is not able to exploit the advantages of evidence combination from different descriptions. Our version of this algorithm differs from Gams in retaining all rule sets and using evidence combination to form overall classifications.

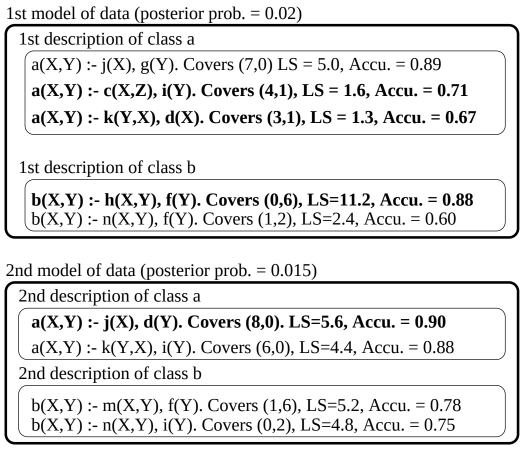


Figure 2. Comparison of evidence combination methods. "Covers (0,6)" for a rule for class 'b' indicates that the body of that rule is true for 0 training examples of class 'a' and for 6 training examples of other class(es). The given accuracy is a Laplace estimate of the accuracy of the rule as estimated from the training data. LS is the degree of logical sufficiency of that rule (explained under "Likelihood Combination" in Section 4).

4. Methods for combining evidence

Our experiments compare four evidence combination methods: Uniform Voting, weighted combination according to Bayesian probability theory (Buntine, 1990), weighted combination according to Distribution Summation (Clark & Boswell, 1991) and Likelihood Combination (Duda *et al.*, 1979). Results using all four evidence combination methods and both learning methods are given in the first appendix. Our goal is to empirically demonstrate that our hypotheses about error reduction apply for a wide variety of evidence combination methods.

Figure 2 shows a situation which will be used to explain the evidence combination methods. Assume for the moment that only the first model has been learned. The rules in bold typeface indicate the rules that have been satisfied for the current test example. The figure indicates that the preconditions of two rules for class a were satisfied by the test example. The first of these rules covers four training examples of class a. The figure also indicates that the second rule of the description of class b covers one training example of class a and two of class b.

4.1. Evidence combination within one description

Before describing evidence combination between descriptions of a given class, we explain how classification occurs when only one model has been learned. Each evidence combination method uses its own kind of *reliability measure*. When only

Table 2. Four evidence combination methods. The composite evidence for a class is obtained by summing the degrees of belief for that class over descriptions of that class.

Evidence comb. method	Descr. 1 class a	Descr. 2 class a	Composite class a	Descr. 1 class b	Descr. 2 class b	Composite class b
Uniform Voting	1	1	2	1	0	1
Bayesian Comb.	0.02 * 0.71 = 0.0142	0.015 * 0.90 = 0.0135	0.0277	0.02 * 0.88 = 0.0176	0.015 * 0 = 0	0.0176
Distribution Sum.	(4,1) + (3,1) = (7,2)	(8,0)	(15,2)	(0,6)	(0,0)	(0,6)
Likelihood Comb.	1.6	5.6	$1.6 \times 5.6 \times 1.75 = 15.68$	11.2	1	$11.2 \times 1 \times 0.57 = 6.384$

one model is being used, and more than one rule in a class description has been satisfied, 3 of the 4 methods described here use only the most reliable of those satisfied rules. We will refer to this as the "single, most reliable rule" bias. See Torgo (1993) for empirical support for using this bias within each rule set.[3] Only the Distribution Summation method takes all satisfied rules within a class description into account. For each method, if the example does not satisfy any rule of any class, each method predicts the class that was most frequent in the training set.

- Uniform Voting - Uniform Voting assigns a uniform reliability of 1 to each rule. It assigns a score of 1 to a class if any rule in that class was satisfied by the test example. Otherwise the score is 0. So this means that for Figure 2, both classes get a score of 1. Uniform Voting then randomly chooses between the classes with the highest scores. Uniform Voting is not competitive with the other methods when using just a single model but it is competitive once several models are used.

- Bayesian Combination (Buntine, 1990) - In Bayesian Combination, there are weights associated with models (the posterior probability of the model) and weights associated with rules (the accuracy of the rule). When only 1 model is being used, only the rule-weights are relevant. The accuracy of a rule r of class i with respect to a set S of examples is the ratio of the number of examples of class i in set S which satisfy the rule divided by the total number of examples (of any class) in S which satisfy the rule. The accuracy is denoted as $p(Class_i|r)$.

  A word about estimation of rule accuracy: we use the training set as set S. This will typically over-estimate the accuracy of the rule so as a correction we use the Laplace estimate (Kruskal & Tanur, 1987). The Laplace estimate of the probability of the event $X = v$ where X is a variable and v is a value which has been observed to occur f times in T consecutive trials is $(f + 1)/(T + k)$ where k denotes the number of possible values that X can take In the context of rule accuracy, using the Laplace estimate means that if N denotes the total number of examples that satisfy rule r and n_i denotes the number of examples of class i that satisfy r, $\frac{n_i+1}{N+2}$ is used as an estimate of the rule accuracy. We use "2" instead of the real number of classes since with respect to rules of class i, all other classes are grouped together as the "negative class."

In Figure 2, the accuracies of the satisfied rules of class a are 0.71 and 0.67 so the more reliable (0.71) is used as the score for class a. Class b only has one satisfied rule so its accuracy (0.88) is used. Bayesian Combination predicts the class with the higher score - class b in this situation.

- Distribution Summation (Clark & Boswell, 1991) - This method associates a k-component vector (the distribution) with each rule. k denotes the number of classes. The vector consists of the numbers of training examples from all k classes covered by that rule. A component-wise sum is formed over all satisfied rules (of all classes) that match a test example to produce a combined vector. So in Figure 2, the distributions of the satisfied rules are added to yield the summed vector: $(4,1)+(3,1)+(0,6)=(7,8)$. Since the highest number in the summed vector corresponds to class b, this method will predict class b.

- Likelihood Combination (Duda *et al.*, 1979) - This method associates the "degree of logical sufficiency of the rule" (LS) (Duda *et al.*, 1979) with each rule. In the context of classification, the LS of a rule of $Class_i$ is defined as the ratio of the following probabilities:

$$\frac{p(rule(\tau) = true \mid \tau \in Class_i)}{p(rule(\tau) = true \mid \tau \notin Class_i)}$$

where τ is a random example. Each probability is estimated using the Laplace method. LS is a generalization of the notion that the body of a rule is completely sufficient to conclude the head of the rule. This method uses the odds form of Bayes rule (Duda *et al.*, 1979) which can be restated for our purposes as:

$$O(Class_i|M_{i1}) = \frac{O(Class_i) \times O(M_{i1}|Class_i)}{O(M_{i1})} \propto O(Class_i) \times O(M_{i1}|Class_i) \tag{1}$$

where M_{i1} denotes the description for class i in the first (and in this section, only) model. The odds of a proposition with probability p are defined to be $p/(1-p)$. In order to calculate $O(M_{i1}|Class_i)$, let $\{R_1,\ldots,R_n\}$ denote the set of rules in description M_{i1} that were satisfied by the example. *If* these rules are conditionally independent given class i, we can write:

$$O(M_{i1}|Class_i) = \prod_j LS_{R_j} \tag{2}$$

where LS_{R_j} is the LS of rule R_j. However, since the rules were learned by a separate and conquer strategy, rather than taking a product of the LS's of the satisfied rules as suggested by Equation 2, it is conceptually (and empirically) better to use only the LS of the most reliable rule.

In Figure 2, class *a* had 14 of the 22 training examples for a "prior" probability of 0.63 and prior odds of 1.75. Class *b* had 8 of the 22 examples for a "prior" probability of 0.36 and prior odds of 0.57. So, in Figure 2, for class *a*, we multiply the prior odds of class *a* with the LS of the more reliable of the two satisfied rules (1.6) to yield a score of 2.8. This score represents the posterior odds of class *a*. Class *b* has prior odds of 0.57 and the LS of the most reliable satisfied rule of class *b* is 11.2, so its score is 6.384. Therefore, Likelihood Combination predicts class *b* in this situation.

We chose Uniform Voting as a "straw man" method which the other methods should be able to beat in terms of accuracy. We chose Bayesian Combination because it is an approximation to the optimal Bayes approach. Distribution Summation was chosen because rules that cover more examples are given higher weight in this method. As Muggleton *et al.* (1992) have noticed, training coverage of a rule is more closely correlated with its test-set accuracy than is its training accuracy. Finally, we chose Likelihood Combination because the logical sufficiency measure used by that method has the flavor of measuring both coverage and accuracy. Most of the rules learned by HYDRA cover no negative training examples. Under these conditions, the Laplace estimate of training set accuracy ranks rules in order of the number of positive examples covered whereas training set LS ranks rules in order of the fraction of positive space covered. Accordingly, we find that rules of minor classes are given relatively higher weights under the LS scheme.

4.2. Evidence combination between descriptions

When more than one model has been learned, classification proceeds by combining evidence for each class from all its descriptions and then finally comparing that degree of evidence to those of the other classes.

- Uniform Voting - In the context of multiple models (Table 2) this method simply counts, for each class, the number of descriptions of that class that have at least one satisfied rule. So, in Figure 2, class *a* gets a score of 2 and class *b* gets a score of 1 so this method predicts class *a*.

- Bayesian Combination - In the general form of Bayesian Combination, the test example, x, should be assigned to the class, c with the highest expected posterior probability:

$$E_{\mathcal{T}}(p(c|x,\vec{x},\vec{c})) = \sum_{T in \mathcal{T}} p(c|x,T)p(T|\vec{x},\vec{c}) \tag{3}$$

where the expectation is taken with respect to $\mathcal{T}$, the model (hypothesis) space of all possible models, $\vec{x}$ denotes the training examples and $\vec{c}$ denotes the class labels of those training examples. x denotes the current test example. $p(c|x,T)$ is the probability of class c given a test example x and a particular model T. $p(c|x,T)$ can be thought of as the degree to which T endorses class c for example x. In this paper, since we are using a "single, most reliable rule" bias, the Laplace accuracy of the most reliable satisfied rule is used for $p(c|x,T)$. $p(T|\vec{x},\vec{c})$ denotes the posterior probability of the model. Briefly, models whose class descriptions are syntactically-compact and are well able to separate the training examples of different classes end up with higher posterior probabilities. Appendix 2 and (Ali & Pazzani, 1995) detail how Buntine's form for the posterior probability of a decision tree (Buntine, 1990) is adapted for the kinds of models described in this paper.

The general Bayesian method is used in our "single, most reliable rule" framework as follows. As Figure 2 indicates, the first model has posterior probability 0.02 and the satisfied rule of class a with highest accuracy has accuracy 0.71. The second model has posterior probability 0.015 and the accuracy of the matching rule is 0.90. This yields an expected posterior probability for class a of 0.0277 ($0.02 * 0.71 + 0.015 * 0.90$). Doing the same for class b yields a degree of belief of 0.0117 for class b. Hence, the test example is assigned to class a.

- Distribution Summation - This method is simply extended to multiple models by doing a vector summation of the distributions of all satisfied rules across all models. So, in Figure 2, this produces an summed vector of $(4,1)+(3,1)+(0,6)+(8,0)+(1,6)=(16,14)$ and consequently the example is assigned to class a.

- Likelihood Combination - In extending this method to multiple models, the prior odds of the class only appear once: let M_i denote the set of class descriptions for class i and M_{ij} denote one such class description. Then the posterior odds of $Class_i$ are given by:

$$O(Class_i|M_i) \propto O(Class_i) \times \prod_j O(Class_i|M_{ij})$$

For the term $O(Class_i|M_{ij})$ we use the LS of the most reliable satisfied rule in M_{ij}. As Table 2 shows, the posterior odds of class a are obtained by multiplying the prior odds (1.75) by the LS of the most reliable matching rule in the first description of class a with the LS of the most reliable matching rule in the second description of class a. This yields posterior odds of 15.68 for class a and posterior odds of 6.384 for class b. Therefore, this evidence combination method will assign the example to class a.

5. Empirical Analyses

For our experiments we chose domains from the UCI repository of machine learning databases (Murphy & Aha, 1992) ensuring that at least one domain from each of the major groups (molecular biology, medical diagnosis ...) was chosen. These include molecular-biology domains (2), medical diagnosis domains (7), relational[4] domains (6 variants of the King-Rook-King (KRK) domain, Muggleton *et al.*, 1989), a chess domain with a "small disjuncts problem" (KRKP; Holte *et al.*, 1989), and attribute-value domains (4 LED variants and the tic-tac-toe problem).

For most of the domains tested here, we used thirty independent trials, each time training on two-thirds of the data and testing on the remaining one-third. The exceptions to this are the DNA promoters domain for which leave-one-out testing has traditionally been used and we follow this tradition to allow comparability with other work. Other exceptions are trials involving the King-Rook-King domain. For this domain, the training and test sets are independently drawn (rather than being mutually exclusive) from the set of all 8^6 board configurations. There is little chance of overlap between training and test sets at the sample sizes we use. Whenever possible we tried to test learned models on noise-free examples (including noisy variants of the KRK and LED domains) but for the natural domains we tested on possibly noisy examples. The large variant of the Soybean data set was used and the 5-class variant of the Heart data set was used.

5.1. Does using multiple rule sets lead to lower error?

In this section we present results of an experiment designed to answer the first of the questions listed in Section 1:

> What effect does using multiple descriptions per class have on classification error as compared to the error produced by using a single description per class?

For this experiment, the Stochastic and Partition methods were used to learn eleven models (we chose an odd number to prevent ties from occurring for the Uniform Voting combination method for two-class domains). Although most of the results in the following sections are given for eleven models, we also performed experiments using one, two and five models. Figure 3 shows the effect of varying the number of models on classification accuracy. Eleven models were used since preliminary experiments indicated that for most data sets, using more than eleven models yields little gains but costs a lot in terms of computation time. Unfortunately, we do not have a method that will indicate the optimal number of models to learn in a given data set.

For the Stochastic method, all literals that had gain at least 0.8 ($\beta = 0.8$) as large as that of the best literal were retained (see Section 5.6 for results on the effect of varying the bucket size). For the Partition method, k was set to the same number of models (eleven) as used during stochastic hill-climbing. For each description

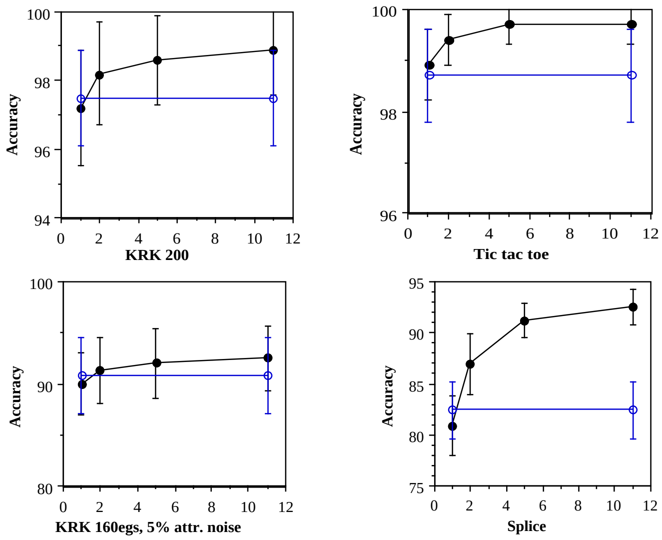


Figure 3. The figures above illustrate the effect of varying number of stochastically-learned models (combined with Uniform-Voting). Open circles represent accuracies obtained by HYDRA, closed circles represent the multiple models method. At least one data set from each of the major types of data sets used in this paper is represented.

generation method, we tested all four evidence combination schemes. The results of using Likelihood Combination on stochastically-generated descriptions are presented in Table 3. Results using all four evidence combination methods and both learning methods are presented in the appendix.

Table 3 compares the accuracies obtained by using a single deterministically learned description to the accuracies obtained by using eleven descriptions. The first column indicates the domain name. Trailing suffixes indicate number of irrelevant attributes (i), number of training examples (e), percentage of attribute noise (a) or percentage of class noise[5] (c). The second column indicates the accuracy that would be attained by guessing the most frequent class. An asterisk signifies that the accuracy of the single description method was not significantly better than guessing the most frequent class. The third column indicates the accuracy obtained by using HYDRA (using information gain) to deterministically learn a single description. The next two columns indicate error ratios for stochastic and k-partition learning respectively. A '+' indicates a significant (using the paired 2-tailed t-test at the 95% confidence level) reduction in error, a '–' indicates a significant increase. For the DNA domain, the t-test is not applicable because we used leave-one-out testing. For this domain, we used a sign-test (DeGroot, 1986).

The data sets are grouped as follows: the first group contains noise-free training data from artificial concepts (for which we know the true class descriptions), the second group contains noisy data from artificial concepts the third contains data

Table 3. Comparison of errors produced by a single description versus two methods (stochastic hill-climbing and k-fold partition learning) of learning multiple descriptions. Eleven models were used and the Likelihood Combination method was used for evidence combination. A '+' indicates that the accuracy of multiple models was significantly higher than that of the single model. A '-' indicates the accuracy was significant lower. An asterisk indicates that the accuracy of the single model version was not significantly better than guessing the most frequent class.

Domain	Default Accuracy	Single Description Accuracy	11 Stochastic Descriptions Accuracy	11 Stochastic Descriptions Error Ratio	11 Partition Descriptions Error Ratio	Number of training examples
Led 8i	10.0%	87.2%	+ 96.4%	+ .28	+ .37	30
Led 17i	10.0%	83.7%	+ 94.6%	+ .33	+ .46	30
Tic-tac-toe	65.3%	99.0%	+ 99.8%	+ .22	+ .38	670
Krkp	52.0%	94.5%	+ 95.5%	+ .82	+ .86	200
Krk 100e	66.7%	95.1%	95.6%	.90	.89	100
Krk 200e	66.7%	98.3%	98.9%	.66	.69	200
Krk 160e 5a	66.7%	91.9%	93.2%	.84	.80	160
Krk 320e 5a	66.7%	94.8%	95.8%	.80	+ .66	320
Krk 160e 20c	66.7%	89.6%	91.1%	.86	.83	160
Krk 320e 20c	66.7%	92.5%	93.4%	.88	+ .79	320
Led 20a	10.0%	94.3%	94.6%	.94	.82	50
Led 40a	10.0%	85.0%	87.7%	.82	.89	50
DNA	50.0%	67.9%	+ 86.8%	+ .41	+ .44	105
Splice	53.4%	85.3%	+ 92.5%	+ .51	+ .62	200
Mushroom	50.0%	97.4%	96.8%	1.24	.96	100
Hypothyroid	90.0%	95.3%	+ 97.8%	+ .47	+ .53	200
BC-Wisconsin	65.5%	93.5%	+ 96.1%	+ .60	+ .70	200
Voting	62.0%	93.5%	+ 94.9%	+ .78	+ .84	100
Wine	39.8%	93.3%	+ 98.9%	+ .16	+ .53	118
Iris	33.3%	91.4%	92.4%	.88	.96	50
Soybean	14.6%	88.5%	+ 92.3%	+ .67	+ .74	288
Horse-colic	63.4%	83.2%	+ 87.1%	+ .77	+ .73	245
Hepatitis	*79.6%	78.8%	79.2%	.98	.99	103
Lymph.	54.7%	77.9%	+ 83.9%	+ .73	.90	110
Audiology	25.3%	72.1%	+ 80.5%	+ .70	+ .76	150
Diabetes	65.1%	71.9%	+ 73.9%	+ .93	+ .93	200
B.Cancer	*70.2%	69.9%	– 67.2%	– 1.09	1.05	190
Heart	*54.1%	54.3%	55.2%	.98	.98	200
Primary-tumor	24.7%	38.8%	38.2%	1.01	– 1.04	225

sets from molecular biology domains and the final group contains probably noisy data from medical diagnosis and other "real world" domains. The domains in the last group are sorted so that those with the highest single model accuracies appear first.

Table 3 shows that stochastic search using Likelihood Combination is able to statistically significantly (95% confidence) reduce or maintain error on all domains except the (Ljubljana) breast-cancer domain. On that breast cancer data set few learning methods have been able to get an accuracy significantly higher than that obtained by guessing the most frequent class suggesting it lacks the attributes relevant for discriminating the classes. The table shows that for approximately half the data sets, error is reduced by a statistically significant margin when using models learned by stochastic search and combined with Likelihood Combination. The appendix shows that the other evidence combination methods and learning methods also lead to statistically significant error reductions for many data sets. There is no significant change in error for most of the other data sets - on very few occasions does the multiple models approach lead to a significant increase in error.

Another striking aspect of the results presented in Table 3 is that the error is reduced by a factor of 6 for the wine data set (representing an increase in accuracy from 93.3% to 98.9%!) and by large (around 3 or 4) factors for LED and Tic-tac-toe. The molecular biology data sets also experienced significant reduction with the error being halved (for DNA this represented an increase in accuracy from 67.9% to 86.8%!). The error reduction is least for the noisy KRK and LED data sets and for the presumably noisy medical diagnosis data sets. Eighty percent of the data sets which scored unimpressive error ratios (above 0.8) were noisy data sets. This finding is further explored in Section 5.4 in which we explore the effect of class noise on error ratios. The fact that the best error ratios were obtained on the noise-free and molecular biology data sets holds for all four of the evidence combinations schemes we used and both description generation methods (see appendix).

The LED domain, in particular, gives us some insight into the effect of irrelevant attributes and class noise on error ratios. As the table shows, learning multiple descriptions helps a lot in reducing errors of the LED data sets with irrelevant attributes. For eight irrelevant attributes, the error is reduced from 12.8% to just 3.6%. This suggests that when irrelevant attributes are present, using multiple descriptions provides a substantial benefit. Backing up this hypothesis are also the DNA and Splice domains for which the error is reduced by a large factor. These domains have many (57 for DNA, 60 for Splice) attributes some of which are probably irrelevant. These observations led us to more carefully investigate the effect of irrelevant attributes on error ratio. The results of those investigations are presented in Section 5.5.

Although error ratios for the noisy data sets represent a statistically significant reduction in error, the ratios are not as impressive as they are for noise-free domains containing irrelevant attributes. Again, the LED data sets provide some insight. The LED variants presented in the table differ by two dimensions: the variants with irrelevant attributes have no noise and the noisy variants have no irrelevant attributes. The LED results suggest that the error ratios obtained through the use of multiple descriptions become less beneficial as the amount of noise increases. This issue is explored in detail in Section 5.4.

In summary, the answer to the question for this section ("What effect does the use of multiple descriptions have on classification error?") is that the use of multiple descriptions leads to significant reductions in classification error for about half of the data sets tested here. For most of the other data sets, the error does not change significantly. Therefore, most of the time, the multiple descriptions approach helps significantly or does not hurt. This is true for both description generation methods and all four evidence combination methods tried here. The table in the appendix presents results for the other generation methods and evidence combination methods.

5.2. Link between error reduction and correlated errors

In this section we explore the following question:

> What is the relationship between the amount of observed error reduction (as measured by error ratio) and the tendency of the learned models to make correlated errors?

Hansen & Salamon (1990) first introduced the hypothesis that the ensemble of models is most useful when its member models make errors totally independently with respect to every other model in the ensemble. They proved that when all the models have the same error and that error is less than 0.5 and they make errors completely independently that the expected ensemble error must decrease monotonically with the number of models. The question we explore here is more general firstly because it does not assume that the errors are made completely independently and secondly because it attempts to explain the *amount* of error reduction in terms of the fraction of correlated errors (ϕ_e).

Now we present a precise instantiation of the concept: "the degree to which the errors made by models of the ensemble are correlated." In our approach, we will compute a correlation for each pair of models in the ensemble $\mathcal{F} = \{\hat{f}_1 ... \hat{f}_T\}$ and ϕ_e will be the average of all those pairwise correlations. Let ϕ_{ij} denote the correlation between the i-th and j-th models. Let $\hat{f}_i(x) = y$ denote the event that model i has classified example x to class y. Let $f(x)$ denote the true class of x. Then ϕ_{ij} has the following definition:

$$\phi_{ij} = p(\hat{f}_i(x) = \hat{f}_j(x), \hat{f}_i(x) \neq f(x))$$

and $\phi_e(\mathcal{F})$, the degree to which the errors in $\mathcal{F}$ are correlated, has the following definition:

$$\phi_e(\mathcal{F}) = \frac{1}{T(T-1)} \sum_{i=1}^{T} \sum_{j \neq i}^{T} p(\hat{f}_i(x) = \hat{f}_j(x), \hat{f}_i(x) \neq f(x))$$

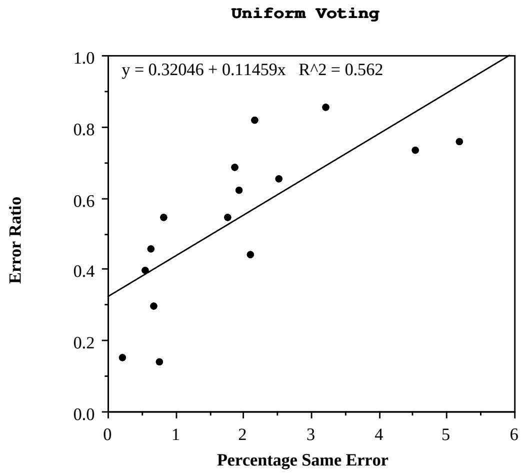


Figure 4. Plot of error ratio as a function of $100 \times \phi_e$. One point represents one data set. Learning method: stochastic hill-climbing, evidence combination method: Uniform Voting).

Figure 3 plots error ratio as a function of percentage of correlated errors ($100 \times \phi_e$) for all domains for which there was a statistically significant reduction in error.[6] The linear correlation coefficient (r) between fraction of correlated errors (ϕ_e) and error ratio (E_r) can be used to measure how well ϕ_e models error *reduction* as measured by E_r. Of the 29 data sets used in this study, significant error reduction was obtained (when using stochastic learning and Uniform Voting) on 15 data sets. Error did not increase significantly for any of the remaining 14 data sets. The r^2 of 0.56 in the Figure shows that 56% of the variance in error ratio can be explained by the tendency of members of the ensemble to make correlated errors. For the other evidence combination methods, the values were 56% (Bayesian Combination), 43% (Distribution Summation) and 41% (Likelihood Combination). When k-fold partition learning was used, the values were 60% (Uniform Voting), 40% (Bayesian Combination), 35% (Distribution Summation) and 41% (Likelihood Combination). This is quite encouraging given that the data sets vary widely in type of class description, optimal Bayes error level, numbers of training examples and numbers of attributes. Another point to note is that ϕ_e is a pairwise measure, whereas what the error rate under Uniform Voting counts is the proportion of the test examples on which at least half of the members in the ensemble make an error.

How stable are these estimates of r^2? In particular, is it possible that we are able to get such a high r^2 simply because of one point luckily appearing near the line of best fit? In order to measure the stability of these estimates of r^2, for each of the eight combinations of learning method and evidence combination method we calculated twenty-nine r^2 values - each time calculating what the r^2 would be if one of the 29 data sets were left out. This analysis (Table 4) shows that the r^2

Table 4. Ranges of leave-one-out estimates of r^2 between error ratio and ϕ_e (the fraction of correlated errors).

	Uniform Voting	Bayesian Combination	Distribution Summation	Likelihood Combination
Stochastic Hill-climbing	[0.54,0.61]	[0.47,0.71]	[0.37, 0.52]	[0.38,0.48]
k-fold Partition Learning	[0.55,0.66]	[0.27,0.56]	[0.28,0.46]	[0.29,0.56]

values presented above do not depend critically on any single data set. We also performed significance tests to compute the likelihood of the observed results under the null hypothesis (that the population correlation, ρ, equals 0). The tests showed that the likelihood of our data given H_0 was less than 0.01 for each of the eight combinations of learning method and evidence combination method. Therefore, we can conclude that there is a significant linear correlation between error ratio and the tendency to make correlated errors for all the learning methods and evidence combination methods used in this study. When ϕ_e is small, multiple models have a substantial impact on reducing error. In Sections 5.4 and 5.5 we investigate how class noise and irrelevant attributes affect ϕ_e and consequently the amount of error reduction achieved by multiple models.

In order to gain insight into why ϕ_e explains so much of the variance in error ratio consider the simpler problem of modeling variation in error within a given data set (this removes possibly confounding variables such as optimal Bayes error rate that vary from one data set to another). Assume that N trials have been conducted to yield N ensemble error values. Assume that the simplest evidence combination method (Uniform Voting) is used and that the data set contains two classes and that the ensemble contains just two models. In this situation, an ensemble error occurs if both the models make an error or if the models disagree and the tie is broken so as to cause an error. Assume that a tie will occur for a negligible proportion of the test examples. Under these assumptions, ϕ_e is an exact measure of ensemble error (E_e).

As ϕ_e is a pairwise measure, how well it models within-dataset ensemble error depends on the size of the ensemble. It is a better model of ensemble errors for ensembles of smaller size. The evidence combination method also affects the ability to model ensemble error using ϕ_e. ϕ_e is a better model of ensemble error obtained by Uniform Voting than it is for evidence combination methods in which different models are given different "voting" weights.

5.3. Gain ties and error reduction

ϕ_e provides a post-hoc way of understanding why the multiple models approach reduces error more for some domains than for other domains. In this section, we explore whether we can approximately *predict* the amount of error reduction due to the use of multiple models. We explore the following question:

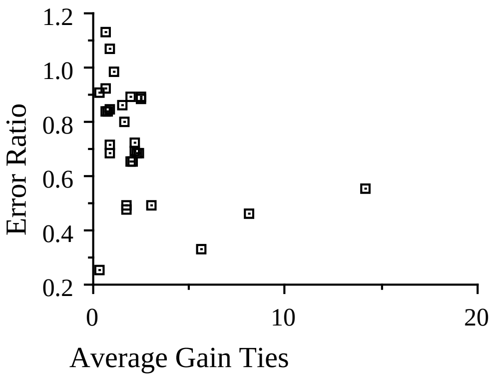
> Can the amount of error reduction observed for a data set be *predicted* from the number of ties in gain experienced by the learning algorithm on that data set?

The motivation for postulating this hypothesis is the observation that each time the stochastic generation method is run, it uses the same training data. However, it is able to generate different descriptions because it randomly picks from the literals whose gain is within some factor β ($\beta \in [0, 1]$) of the gain of the highest literal. If there are many such literals then the possibility for syntactic variation from description to description is greater. The greater syntactic diversity may leads to less correlation of errors as measured by ϕ_e which in turn may lead to lower (i.e. better) error ratios. As a first approximation measure of the amount of syntactic variety in a data set as experienced by a learning algorithm, consider the number of literals that tie for the highest information gain. If n literals tie for gain, that event is recorded as representing $n - 1$ ties in gain. The total number of ties experienced during learning a model is then divided by the number of literals in the model to produce the quantity g, the "average number of gain ties" for that data set. A large number of such ties are a problem for a hill-climbing deterministic learner but represent an opportunity for the multiple model learner. Figure 5 plots error ratio as a function of average gain ties (each point represents results for one data set from Table 3). The figure shows that some of the largest reductions in error are obtained for data sets for which such ties are frequent (on average, there were 5.1 gain ties on the wine data set, 6.6 for the DNA promoters data set and 2.5 for the Splice data set). However, the figure also shows that a high average value for ties in gain is not a *necessary* condition for significant reduction of error. For example, multiple models are able to achieve low error ratios on the Tic-Tac-Toe and the noise-free LED variants (bottom left of figure) even though there are not many ties in gain for those data sets.

In summary, the answer to the question posed in this section is that if the number of gain ties experienced on average for a data set is large (say 2 or more) then that data set will benefit quite a lot (i.e. have its error reduced by at least 40%) from the use of multiple models. In our experiments, we have seen no exceptions to this trend. However, if the number of gain ties is small, the amount of error reduction cannot be predicted. As Figure 5 shows, these gain-ties results are not just true for HYDRA - they are also true for ID3 (Quinlan, 1986) - the canonical decision tree learning algorithm.

5.4. Effect of class noise

The results of Section 5.1 showed that the majority (80%) of data sets for which unimpressive error ratios (above 0.8) were recorded were data sets with significant



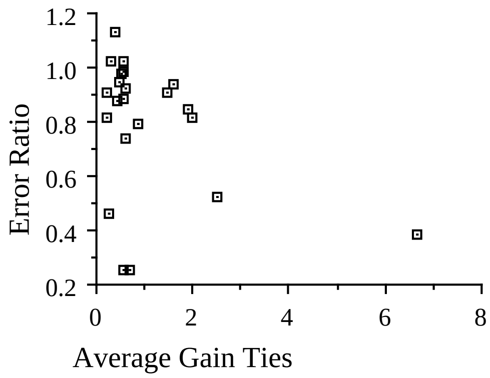


Figure 5. Error ratio as a function of average gain ties for decision trees (left) and rule sets (right). The ensembles of decision trees contained eleven decision trees stochastically learned with respect to the entropy gain function. The ensembles of eleven rule sets were learned using stochastic hill-climbing and combined using Likelihood Combination. Similar plots are obtained for other evidence combination methods and the other learning method.

Table 5. Effect of increasing class noise on error ratios (using 11 stochastically-learned models and Uniform Voting for evidence combination). Similar plots are obtained for other evidence combination methods and the other learning method.

Class Noise	KRK 100 Err. ratio	KRK 100 ϕ_e	TTT 200 Err. ratio	TTT 200 ϕ_e	Wine Err. ratio	Wine ϕ_e	BC-Wisc. Err. ratio	BC-Wisc. ϕ_e
10%	.85	3.6%	.70	3.9%	.23	1.5%	.55	2.7%
20%	.88	5.0%	.77	5.6%	.30	2.5%	.56	3.6%
30%	.98	7.0%	.92	7.4%	.48	4.1%	.62	4.4%
40%	.95	8.9%	.95	8.9%	.53	5.2%	.69	5.5%

amounts of noise. Furthermore, experiments on the LED domain provided preliminary evidence that the addition of attribute noise increases (worsens) error ratios. In this section we follow up on that hypothesis by asking:

> How does increasing the amount of class noise affect the amount of error reduction?

We choose to study the effect of class noise rather than attribute noise because attributes in some domains have more values than attributes in other domains and an attribute with fewer values is more likely by chance to have large information gain. Therefore it would not be easy to compare levels of attribute noise across domains.

Table 5 shows the effect of adding class noise to four very different kinds of data sets. Noise was only added to the training data. We chose the wine and tic-tac-toe

Table 6. Distribution of ensemble errors as a function of the number of models correctly classifying a test example. Learning method: stochastic hill-climbing; evidence combination method: Uniform Voting. Eleven models were combined using Uniform Voting so an ensemble error occurs if five or fewer of the models made an error. Values at 20% and 30% noise lie in between the values presented in the table.

Number of models got test eg. correct	BC. Wisc. 10% noise	BC. Wisc. 10% noise	BC. Wisc. 40% noise	BC. Wisc. 40% noise
	% of ensemble errors	Cumulative % of ensemble errors	% of ensemble errors	Cumulative % of ensemble errors
0	15.4%	15.4%	27.6%	27.6%
1	7.7%	23.1%	11.2%	38.8%
2	20.5%	43.6%	12.2%	51.0%
3	7.7%	51.3%	18.4%	69.4%
4	15.4%	66.7%	14.3%	83.7%
5	33.3%	100.0%	16.3%	100.0%

data sets because the multiple models approach was able to reduce error by a large amount (error ratios of 0.16 and 0.22 respectively) for these data sets. We wanted to see if this advantage would be eroded by the addition of noise. The table shows that for each of the four chosen data sets the advantage yielded by the multiple models approach lessens as class noise is increased.

More careful examination of the patterns of errors of models in the ensemble shows that at 40% noise, a relatively larger proportion of the test examples on which the ensemble made an error were incorrectly classified by *all* the models in the ensemble. That is, as noise increases, some of the examples become "hard" for all the models.

In a follow-up experiment (Table 6), we studied the distribution of the ensemble errors. We wanted to know what proportion of the ensemble errors were caused by all the models making an error and what proportion were caused by a narrow majority of models making an error. The first column indicates the number of models that correctly classified the test example. The remaining columns are arranged in two groups. Columns two and three present results for 10% class noise, the last two columns present results for 40% noise. The i-th row corresponds to test examples that were *correctly* classified by i (out of 11) models. The first column in each set indicates the number of test examples characterized by that situation. Let a m/n split indicate the situation for a test example where m models make a correct classification and n make a mistake ($m + n = 11$). Therefore, the table indicates that a 0/11 split occurred on 15.4% of the test examples after learning with 10% class noise and it occurred on 27.6% of the test examples after learning with 40% class noise. Therefore, the table indicates that that as noise level increases, all the models misclassify a test example on a *greater proportion* of the test examples for which an ensemble error is made. This indicates that as noise level increases, some test examples become more difficult for all the models.

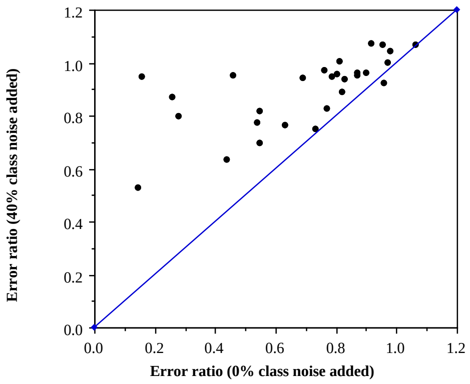


Figure 6. Comparison of error ratios at 0% added class noise and 40% added noise. Learning method: stochastic hill-climbing; evidence combination method: Uniform Voting.

Figure 6 compares the error ratio (11 models, stochastic learning, Uniform Voting) with 0% added noise to that with 40% added noise. In each case, the addition of noise causes the error ratio to go towards 1 indicating the erosion of the advantage of the multiple models approach.

In summary, the answer to the question of this section ("How does increasing the amount of class noise affect the amount of error reduction?") is that increasing class noise causes the multiple models approach to produce poorer error ratios. Extrapolation of these results suggests that at 100% noise the error ratios for all data sets would be 1.0. This makes sense because the training data contains no discrimination information so there is no reason to expect the multiple models approach to do better than the single models approach.

5.5. Effect of irrelevant attributes

The experiments presented in Section 5.1 provide preliminary evidence that the benefit of using the multiple models approach increases with increasing numbers of irrelevant attributes. In this section we describe further experiments to explore this question:

> How does increasing the number of irrelevant attributes affect the amount of error reduction?

To study this question, we added varying number of Boolean irrelevant attributes to a representative sample of data sets. We chose Boolean attributes rather than constructing irrelevant attributes whose values were domain specific because the attributes in some data sets can take on many more values than attributes in other data sets leading to comparison difficulties.

Table 7. Error ratio as a function of number of added Boolean irrelevant attributes (using Uniform Voting of eleven stochastically generated models). The number below each data set identifier indicates the number of training examples. "5a" indicates 5% attribute noise. Similar results are obtained for other evidence combination methods and the other learning method.

Number of irrelevant attributes	KRK 100 Error Ratio	KRK 100 Avge. gain ties	KRK 5a 160 Error Ratio	KRK 5a 160 Avge. gain ties	Splice 200 Error Ratio	Splice 200 Avge. gain ties	BC. Wisc. 200 Error Ratio	BC. Wisc. 200 Avge. gain ties	Wine 118 Error Ratio	Wine 118 Avge. gain ties
0	0.85	0.42	0.81	0.54	0.44	2.51	0.55	0.85	0.13	5.03
3	0.73	0.47	0.67	0.58	0.42	2.78	0.53	0.45	0.13	6.61
20	0.66	0.55	0.64	0.59	0.39	2.93	0.45	0.67	0.11	23.1
50	0.52	0.96	0.55	1.00	0.38	2.81	0.41	1.19	0.11	104.2

Table 7 corroborates the hypothesis that the multiple models approach is able to attain especially impressive error reductions when many irrelevant attributes are present in the data. The table shows that error ratio decreases as a function of increasing numbers of irrelevant attributes. To understand this, consider the Uniform Voting evidence combination scheme. For the multiple models approach to make an error due to irrelevant attributes, at least half of the learned models need to involve an irrelevant attribute that leads to a classification error. If the number of irrelevant attributes is not too large, it is unlikely that at least half of the models will be affected in this manner. Therefore, the multiple models approach will not make an error in this situation. But the single model approach need only make a mistake due to learning a rule involving an irrelevant attribute early in its separate and conquer strategy for most of the subsequent rules to go off track. Hence the single model approach is much more likely to suffer due to irrelevant attributes. Figure 7 extends the irrelevant attributes experiment to all 29 data sets. It plots the error ratio obtained after the addition of 50 irrelevant binary attributes against the error ratio before the addition of any irrelevant attributes. The fact that most of the plotted points lie below the diagonal indicate that for most of the data sets adding irrelevant attributes leads to smaller (better) error ratios.

Table 7 also shows that the average number of gain ties experienced increases as the number of irrelevant attributes increases. This confirms the results (Section 5.3, Figure 5) that better (lower) error ratios are obtainable for data sets where the learning algorithm experiences more gain ties.

Consider, however, what would happen if an arbitrarily large number of irrelevant attributes were to be added to a data set. By adding enough irrelevant attributes, one could force all the learned models to go astray. In this situation, one would expect that the error ratio should go to 1 as both the deterministic and multiple models approaches would perform at chance level. Hence, we predict that for large enough numbers of irrelevant attributes the error ratio would *increase* with increasing numbers of irrelevant attributes. This hypothesis is difficult to test with data sets of reasonable size because such data sets tend to have literals with high information gain and one needs exponentially many irrelevant attributes for an irrelevant attribute to have higher information gain purely by chance for reasonably-

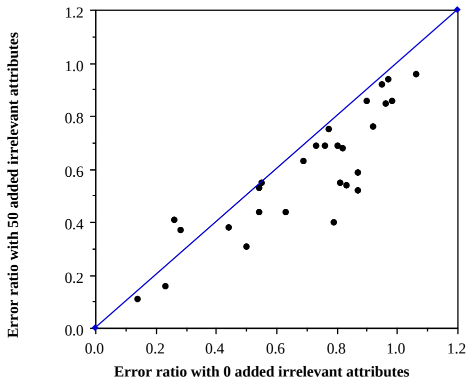


Figure 7. Comparison of error ratios with 0 added irrelevant binary attributes and 50 added irrelevant binary attributes. Learning method: stochastic hill-climbing; evidence combination method: Uniform Voting.

Table 8. Error ratio as a function of number of added Boolean irrelevant attributes for *small* sample sizes (using ensembles of eleven stochastically-learned models; combined with Uniform Voting). The number below each data set identifier indicates the number of training examples.

Number of irrel. attributes	KRK 20	KRK 5% attr. 20	Splice 20	BC. Wisconsin 20	Wine 20
0	1.00	0.98	1.00	0.68	0.44
3	0.99	1.06	0.91	0.67	0.45
20	1.08	1.01	0.92	0.68	0.59
50	0.95	0.97	0.92	0.65	0.64

sized data sets. So, to test this hypothesis, we performed 100 trials with training sets of size 20. In particular, we were interested to see if the exceptionally low error ratio obtained on the wine data set could be made to increase with increasing numbers of irrelevant attributes. Table 8 shows that for very small training set sizes, adding irrelevant attributes makes no significant difference to error ratios in 4 domains and *increases* the error for the wine data set thus validating our hypothesis.

In summary, the answer to the question posed in this section ("How does increasing the number of irrelevant attributes affect the amount of error reduction?") is that error ratios initially decrease as irrelevant attributes are added thus providing an opportunity for the multiple models approach. However, beyond some point, adding irrelevant attributes will begin to hurt the multiple models approach and error ratios will begin *increasing* towards 1. In the limit, neither the single model

Table 9. Effect of varying diversity on the tendency to make correlated errors (ϕ_e) and accuracy (learning method: stochastic hill-climbing; evidence combination method: Uniform Voting).

Domain	Accuracies				ϕ_e			
	Bucket size 4	Bucket size 6	Bucket size 8	Bucket size 20	Bucket size 4	Bucket size 6	Bucket size 8	Bucket size 20
LED-8	95.1	95.1	93.1	92.3	0.62%	0.62%	0.68%	0.71%
KRK 100	93.9	93.1	93.0	92.1	1.10%	2.46%	2.55%	2.94%
Iris	94.4	94.3	94.5	94.5	1.93%	1.74%	1.59%	1.57%
Diabetes	73.9	74.3	74.1	74.2	8.09%	8.01%	7.95%	7.81%
Splice	93.3	92.9	92.9	90.9	1.38%	1.50%	1.60%	2.01%

approach or the multiple models approach will be much use, and the error ratio will be 1.

5.6. Effect of diversity

In this section we explore the following question:

> Does increasing the diversity of the models *necessarily* lead to greater reduction in error?

This question is motivated by the conclusions in Kwok & Carter (1990) in which they show (on two domains) that ensembles consisting of syntactically more diverse decision trees are able to achieve lower error rates than ensembles consisting of less diverse decision trees.

In this experiment, we modified the stochastic hill-climbing algorithm slightly by allowing the user to specify a fixed bucket size. Larger bucket sizes lead to ensembles whose members are more syntactically diverse. We chose a variety of domains for this study: LED-8 and KRK 100 are noise-free, Diabetes and Iris may contain class and attribute noise and the Splice domain may contain classes which can be succintly described with "m of n" rules (e.g. Spackman, 1988). Table 9 shows the accuracies obtained by combining eleven stochastically generated models using the Uniform Voting evidence combination method. Our hope is that increasing the bucket size will lead to an increase in ensemble accuracy. However, as Table 9 shows, increasing the bucket size does not always lead to an increase in ensemble accuracy. To achieve higher accuracy, the models should be diverse *and* each model must be quite accurate. In fact, it is easy to produce uncorrelated errors by learning less accurate models.

A more detailed examination of the results shows that many equally accurate models were learned for the Iris, Diabetes and Splice domains by increasing the bucket size. But for the noise-free, artificial concept data sets (Led-8 and "KRK 100") increasing the bucket size led to a few accurate models and many less accurate models. For LED and KRK, we know the target definitions so we know that all the relevant attributes are presented to the learning algorithm. Maybe for these data sets, all the very accurate models that can be learned are syntactically similar so increasing syntactic diversity is not a good idea in this kind of data set. This experiment suggests that although theory prescribes evidence combination from all models in the model or hypothesis space (Buntine, 1990), in practice only a small number of models are learned and so it may be necessary to screen out less accurate models in order to maximize overall accuracy.

To summarize, our experiments indicate that in order to minimize ensemble error, it is necessary to balance increased diversity with competence - ensuring the diverse members of the ensemble are all competent (accurate). The "hold-back" approach would seem to be an obvious approach. However, for some of the small data sets presented here, using a hold-back set may decrease accuracy since there would not be enough examples to learn good models.

6. Previous work

Breiman (in press; 1994) provides a characterization of learning algorithms which are amenable to the multiple models approach. He puts forward the notion of an "unstable" algorithm - an algorithm for which small perturbations in the training set will lead to significant differences in predicted classifications on an independent (test) set of examples. Breiman shows that decision-tree induction algorithms and neural-network algorithms are unstable whereas the basic nearest-neighbor algorithm is not. This work differs from ours in that we provide a characterization of domains for which the multiple models approach will be beneficial (many irrelevant attributes, low noise levels) whereas Breiman characterizes the learning algorithm.

Schapire's Boosting algorithm (Schapire, 1990) is the only learning algorithm which explicitly attempts to learn models that make errors statistically independently. Boosting learns from an on-line "stream" of examples. Subsequent models are constructed on training sets that amplify the number of examples misclassified by earlier models. The idea is to concentrate on the difficult examples. However, Schapire's method could not be used to learn many models on the modest training set sizes used in this paper because the number of training examples required rapidly increases as a function of the accuracy of earlier models. Modified-boosting (Freund & Schapire, 1995) designed to work with small data sets has not been proved empirically and may end up concentrating noisy examples in subsequent training sets.

The only previous work involving learning *relational* multiple models (apart from our own, Ali & Pazzani, 1995) has been done by Kovacic (1994). Kovacic shows that learning multiple models by running mFOIL (Dzeroski, 1992) several times using simulated annealing yields significantly lower error rates than mFOIL on the KRK and Finite-element mesh data sets.

Previous work related to the effect of noise and multiple models includes that of Kovacic (1994) and Gams (1990). Our observation that error ratios asymptote to 1 as (class) noise is added is consistent with results tabulated in (Kovacic, 1994) and (Gams, 1990) although those authors did not explore the issue in detail as they did not attempt to explain the variation in error reduction from one domain to another.

Previous work on diversity and multiple models has been done by Kwok & Carter (1990) in which they showed that allowing the root of a decision tree to vary from model to model produces more diverse and more accurate ensembles than if variation is only allowed further down the tree. Our work builds on this by showing that in some situations one is forced to trade-off diversity for accuracy - in such situations many syntactically-diverse and accurate models may not exist. Buntine (1990) also presents results in which option trees are able to achieve better error rates than ensembles of trees obtained by different way of pruning a single initial tree. He postulates that this is because different prunings do not lead to trees that are as diverse as those captured by the option-tree representation.

7. Conclusions

Our experiments confirmed previous work that using multiple descriptions lowers the generalization error. Because our experiments used a large sample of data sets from the UCI repository we were able to find three data sets (not previously used in multiple models work) for which the multiple models approach offers striking error ratios: 1/7 for wine, 1/5 for tic-tac-toe and 1/2.5 for DNA.

However, multiple models work in ways different to those we had anticipated. In particular, they were better at reducing error on tasks which were already fairly accurate (reduced error for Tic-tac-toe from 1% to 0.2%) than they were at reducing error on noisy domains. Such noisy data sets may be called "data-limiting." However, when the limiting factor is not the noise or difficulty of the data, the multiple models approach provides an excellent way of achieving large reductions in error. One situation in which this occurs is for data sets with many irrelevant attributes. The information necessary to differentiate the classes is present in the data but the deterministic hill-climbing learning algorithm may have difficulty finding it. On such ("search-limiting") data sets, the multiple models approach does increasingly better than the single model as the number of irrelevant attributes is increased. We also find that the average number of gain ties experienced increases as the number of irrelevant attributes increases. This confirms our earlier results that the multiple models approach does especially well when there are many gain ties. Beyond some point, however, adding irrelevant attributes begins to hurt the multiple models approach. In the limit, neither the single model approach or the multiple models approach will be much use, and the error ratio will be 1.

We have shown that there is a substantial (linear) correlation between the amount of error reduction due to the use of multiple models and the degree to which the errors made by individual models are correlated. Therefore, we conclude that a major factor in explaining the variance in error reduction is the tendency of the learned models to make correlated errors. But why is it possible to learn models that do not make correlated errors for some domains and not for others? Part of the answer is that it is possible to learn models that make different kinds of errors for domains for which there are many ties in gain. To follow up on this, we tried to increase the number of gain ties for each data set by adding 50 irrelevant binary attributes to each data set. This increased the number of gain ties experienced and also produced greater reduction in error suggesting that an abundance of gain ties is a problem for the single model hill-climbing learning method but an opportunity for the multiple models approach.

Acknowledgments

This work was supported by NSF grant #IRI-9310413. We also acknowledge three anonymous reviewers, Wray Buntine and Padhraic Smyth for helpful suggestions and Pedro Domingos, Dennis Kibler and Dan Frost for reading preliminary versions of the paper.

Appendix 1

The appendix contains a table of accuracies for all four evidence combination methods crossed with the two multiple model learning methods and the single model, deterministic hill-climbing method.

Appendix 2

The posterior probability of a model, $p(T|\vec{x},\vec{c})$, is computed as follows (this presentation follows that in Buntine, 1990): Using Bayes' rule, we can write:

$$p(T|\vec{x},\vec{c}) \propto p(\vec{x},\vec{c}|T) \times p(T) \tag{2.1}$$

$p(T)$ is the prior probability of the model T. By further assuming that the training examples are independent given the model, we can write:

$$p(\vec{x},\vec{c}|T) = \prod_{i=1}^{N} p(x_i, c_i|T) \tag{2.2}$$

where N denotes the size of the training set. Following Buntine, we assume that we can divide up the training set into subsets which correspond to different types of training examples (these can be different disjuncts or in Buntine's case, different leaves of a decision tree). Let there be V such subsets and let $n_{j,k}$ denote the number of training examples of class j in the k-th subset. Then we can write

$$p(\vec{x},\vec{c}|T) = \prod_{k=1}^{V} \prod_{j=1}^{C} \phi_{j,k}^{n_{j,k}} \tag{2.3}$$

Table 1.1. The table below presents a comparison of methods of generating models and of evidence combination methods. Each model generation method is represented by four columns corresponding to the evidence combination methods. They are, in order: Uniform Voting (U), Bayesian Combination (B), Distribution Summation (D) and Likelihood Combination (L). '+' indicates a significant (95% confidence) increase in accuracy as compared to the single model method; '-' indicates a significant decline.

Task	Deterministic, single model Hill-climbing				Stochastic Hill-climbing				k-fold partition Learning			
	U	B	D	L	U	B	D	L	U	B	D	L
led-8i	85.2	91.7	91.0	89.2	+ 97.0	+ 97.9	+ 97.7	+ 98.0	+ 97.2	+ 97.8	+ 98.0	+ 97.9
led-17i	76.5	86.0	85.0	83.5	+ 95.4	+ 96.3	+ 96.2	+ 96.4	+ 94.9	+ 95.6	+ 95.1	+ 95.5
TTT	98.7	99.0	98.9	99.0	+ 99.7	+ 99.8	+ 99.5	+ 99.8	+ 99.7	+ 99.8	+ 99.5	+ 99.6
krkp	92.5	94.5	94.5	94.5	+ 95.3	95.2	+ 95.4	+ 95.5	+ 95.0	+ 95.4	+ 95.4	+ 95.3
KRK 100e	94.7	95.2	95.2	95.1	95.5	95.6	94.6	95.6	95.5	95.9	94.3	95.7
KRK 200e	97.6	98.3	98.3	98.3	+ 98.8	98.9	98.2	98.9	+ 98.7	98.8	98.1	98.8
KRK 160e 5a	90.8	91.9	92.0	91.9	92.5	92.5	91.8	93.2	+ 93.0	92.4	92.3	93.5
KRK 320e 5a	93.4	94.8	94.8	94.8	+ 94.9	95.0	94.9	95.8	+ 95.8	94.9	95.4	+ 96.6
KRK 160e 20c	88.6	89.6	89.6	89.6	90.3	90.5	90.7	91.1	+ 90.9	90.2	+ 91.4	91.3
KRK 320e 20c	91.7	92.5	92.6	92.5	92.6	92.8	93.0	93.4	+ 93.5	92.8	93.6	+ 94.1
led 20a	92.7	94.3	93.0	94.3	93.7	94.3	93.7	94.7	94.0	94.3	94.3	95.3
led 40a	81.0	85.7	82.0	85.0	84.7	87.0	85.3	87.7	86.0	84.7	85.3	86.7
dna	59.4	67.9	67.9	67.9	+ 86.8	+ 90.6	+ 87.7	+ 86.8	+ 84.0	+ 80.2	+ 85.8	+ 85.8
splice	82.4	85.3	85.3	85.3	+ 92.5	+ 91.1	+ 92.3	+ 92.5	+ 91.0	+ 90.6	+ 91.0	+ 90.9
mushroom	97.4	97.4	97.5	97.4	98.0	98.0	96.8	97.3	98.1	97.4	-95.5	97.5
hypothyroid	97.4	97.4	97.4	95.3	97.8	97.6	97.4	+ 97.8	97.8	97.9	97.7	+ 97.5
wisc	92.5	93.5	93.6	93.5	+ 95.8	+ 95.5	+ 94.9	+ 96.1	+ 95.1	+ 95.1	93.7	+ 95.5
voting	93.1	93.5	93.4	93.5	+ 94.2	94.1	+ 94.6	+ 94.4	+ 94.4	94.1	+ 94.6	+ 94.5
wine	92.3	93.3	93.4	93.3	+ 98.7	+ 98.2	+ 98.5	+ 98.7	+ 97.0	+ 97.1	+ 97.5	+ 96.5
iris	90.2	91.4	91.1	91.4	+ 92.8	92.6	92.2	92.4	90.8	91.5	90.1	91.7
soybean	84.6	88.5	88.5	88.5	+ 91.6	+ 91.6	+ 91.0	+ 92.2	+ 90.8	+ 90.8	+ 90.3	+ 91.5
colic	82.3	83.2	83.3	83.2	+ 86.7	+ 86.0	+ 87.7	+ 87.0	+ 87.2	+ 86.1	+ 86.4	+ 87.8
hepa.	78.8	78.9	78.8	78.8	80.2	78.9	79.4	79.5	79.8	78.9	78.0	79.1
lymph	76.6	78.1	78.5	77.9	+ 83.8	+ 82.6	+ 82.4	+ 83.8	+ 78.9	80.4	80.3	80.1
audio.	71.5	72.1	72.0	72.1	+ 80.5	+ 79.3	+ 78.3	+ 80.3	+ 78.3	+ 78.3	+ 77.0	+ 78.7
diabetes	70.6	72.0	72.1	72.0	73.2	72.8	74.4	73.6	73.4	72.4	74.5	73.8
cancer	69.9	69.9	69.8	69.9	68.6	68.1	70.2	-67.3	69.4	68.5	+ 72.0	68.5
heart	54.2	54.3	54.2	54.3	56.0	55.7	+ 57.6	55.1	+ 56.3	54.9	+ 57.4	55.3
prim.	37.5	38.8	39.2	38.8	38.7	40.3	+ 42.8	38.3	37.3	39.1	41.0	-36.1

where $\phi_{j,k}$ represents the probability of generating a single example of class j in the k-th subset and C denotes the number of classes. One can then show (Buntine, 1990) that the contribution to the posterior from the k-th subset can be modeled by:

$$\frac{B_C(n_{1,k} + \alpha, \ldots, n_{C,k} + \alpha)}{B_C(\alpha, \ldots, \alpha)} \tag{2.4}$$

where B_C is the C-dimensional beta function and α is a parameter which denotes the "weight" (in number of examples) that should be associated with the prior estimate $(1/C)$ of $\phi_{j,k}$: Putting equations 2.3 and 2.4 together, we get:

$$p(\vec{x}, \vec{c}|T) = \prod_{k=1}^{V} \frac{B_C(n_{1,k} + \alpha, \ldots, n_{C,k} + \alpha)}{B_C(\alpha, \ldots, \alpha)} \tag{2.5}$$

Since, $p(\vec{x}, \vec{c}|T)$ can be computed, then using Equation 2.1, the posterior probability, $p(T|\vec{x}, \vec{c})$ can be calculated, so (using Equation 3) the final quantity of interest, the expected posterior probability, can be calculated.

The foregoing discussion is enough to calculate posterior probabilities of models that are decision trees. It depends on the observation that the training examples can be partitioned into V disjoint subsets. We adapt it for the types of models considered in this paper in which a separate description is learned for each class by observing that such a model partitions the training examples C (number of classes) times. This is because each class description partitions all the training examples since each description contains a "default rule" - one whose body is the literal *true*. Then in order to compute the posterior probability of such a model, we simply take the geometric average of the posterior probabilities of all the class descriptions:

$$p(T|\vec{x}, \vec{c}) \propto p(T) \times \left(\prod_{i=1}^{C} \prod_{ij \in R_i} \frac{B(n_{1,ij} + \alpha, n_{2,ij} + \alpha)}{B(\alpha, \alpha)} \right)^{1/C} \tag{2.6}$$

R_i denotes the i-th class description and ij indexes individual rules. Since, within the class description for the i-th class, classes are grouped into two pseudo-classes (class i is called the "positive" class, all the other classes are combined into the "negative" class), we can use $k = 2$ in Equation 2.4 to obtain the Beta function terms in Equation 2.6.

Notes

1. Actually, we use a form of clause that is an extension of a Horn clause since we allow negated literals in the body of clauses and Horn clauses do not.
2. Although FOIL is an algorithm that learns class descriptions consisting of relational (first-order) clauses, in this paper we are not concerned with issues pertaining to relational learning or inductive logic programming (e.g. Dzeroski & Bratko, 1992). We present results on the interaction of inductive logic programming and learning multiple models in (Ali & Pazzani, 1995).

3. Ali & Pazzani (1995) presents details on how to deal with recursive concepts in the "single, most reliable rule" framework.
4. King-Rook-King is a fully determinate domain so it can be converted into attribute-value form as is done by Lavrac & Dzeroski (1992). However, in this paper, that knowledge is not utilized by the learning programs so the domain has to be treated as a relational domain. FOIL and HYDRA can also run on non-determinate domains.
5. $x\%$ class noise means that the class assignments of $x\%$ of the examples were randomly re-assigned - for a two class problem, this means $\frac{x}{2}\%$ of the examples will bear incorrect class labels.
6. The r^2's between E_r and ϕ_e without the significant error reduction restriction are: 50.7% (Uniform), 33.7% (Bayes), 6.8% (Distribution) and 31.6% (Likelihood). The Mushroom data set causes a problem for the Distribution combination strategy because both the ensemble error and multiple models error are close to 0 so the ratio cannot be reliably estimated. The r^2 for Distribution increases to 21.1% without Mushroom.

References

Ali, K., & Pazzani, M. (1992.) Reducing the small disjuncts problem by learning probabilistic concept descriptions. In Petsche, T., Judd, S. & Hanson, S. (Eds.), *Computational Learning Theory and Natural Learning Systems, Vol. 3.* Cambridge, Massachusetts: MIT Press.

Ali, K., & Pazzani, M. (1993.) HYDRA: A Noise-tolerant Relational Concept Learning Algorithm In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence* Chambery, France: Morgan Kaufmann. Ali, K. & Pazzani, M. (1995.) HYDRA-MM: Learning Multiple Descriptions to Improve Classification Accuracy *International Journal on Artificial Intelligence Tools, 1 & 2,* 115-133.

Ali, K., & Pazzani, M. (1995.) Learning Multiple Relational Rule-based Models. In Fisher, D., & Lenz, H. (Eds.), *Learning from Data: Artificial Intelligence and Statistics, Vol. 5.* Fort Lauderdale, FL: Springer-Verlag.

Baxt, W.G. (1992.) Improving the Accuracy of an Artificial Neural Network Using Multiple Differently Trained Networks *Neural Computation, 4,* 772-780.

Brazdil, P., & Torgo, L. (1990.) Knowledge Acquisition via Knowledge Integration In *Current Trends in Knowledge Acquisition* : IOS Press.

Bernardo, J.M. & Smith, A.F.M. (1994.) *Bayesian Theory.* : John Wiley.

Breiman, L. (1994.) *Heuristics of instability in model selection.* (Technical Report University of California, Berkeley). Statistics Department,

. Breiman, L. (in press.) Bagging Predictors *Machine Learning, ?,* ?.

Buntine W. (1990.) *A Theory of Learning Classification Rules.* Doctoral dissertation. School of Computing Science, University of Technology, Sydney, Australia.

Clark, P., & Boswell, R. (1991.) Rule Induction with CN2: Some Recent Improvements In *Proceedings of the European Working Session on Learning, 1991* : Pitman.

Danyluk, A., & Provost, F. (1993.) Small Disjuncts in Action: Learning to Diagnose Errors in the Local Loop of the Telephone Network In *Proceedings of the Tenth International Conference on Machine Learning* Amherst, MA: Morgan Kaufmann.

De Groot M.H. (1986.) *Probability and Statistics.* Reading,MA: Addison-Wesley.

Drobnic, M. & Gams, M. (1992.) Analysis of Classification with Two Classifiers. In B. du Boulay and V.Sgurev, *Artificial Intelligence 5: Methodology, Systems, and Applications.* : North-Holland.

Drobnic, M. & Gams, M. (1993.) Multistrategy Learning: An Analytical Approach In *Proc. 2nd Intern. Workshop on Multistrategy Learning* Harpers Ferry, WV: ?.

Drucker, H., Cortes, C., Jackel, L., LeCun, Y. & Vapnik V. (1994.) Boosting and Other Machine Learning Algorithms In *Machine Learning: Proceedings of the Eleventh International Conference* New Brunswick, NJ: Morgan Kaufmann.

Duda, R., Gaschnig, J., & Hart, P. (1979.) Model design in the Prospector consultant system for mineral exploration. In D. Michie (ed.), *Expert systems in the micro-electronic age.* Edinburgh, England: Edinburgh University Press.

Freund, Y. & Schapire, R.E. (1995.) A Decision-Theoretic Generalization of On-Line Learning and an application to Boosting. In Vitanyi, P. (Ed.), *Lecture Notes in Artificial Intelligence, Vol. 904.* Berlin, Germany: Springer-Verlag.

Dzeroski, S., & Bratko, (1992.) Handling noise in Inductive Logic Programming In *Proceedings of the International Workshop on Inductive Logic Programming* Tokyo, Japan: ICOT Press.

Gams, M., & Petkovsek, M. (1988.) Learning From Examples in the Presence of Noise In *8th International Workshop; Expert Systems and their applications, Vol. 2* Avignon, France: .

Gams, M. (1989.) New Measurements Highlight the Importance of Redundant Knowledge In *European Working Session on Learning (4th: 1989)* Montpeiller, France: Pitman.

Hansen, L.K, & Salamon, P. (1990.) Neural Network Ensembles *IEEE Transactions on Pattern Analysis and Machine Intelligence, 12, 10,* 993-1001.

Holte, R., Acker, L., & Porter, B. (1989.) Concept Learning and the Problem of Small Disjuncts In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence* Detroit, MI: Morgan Kaufmann.

Howell, D. (1987.) *Statistical Methods for Psychology.* Boston, MA: Duxbury Press.

Kong, E.B., & Dietterich, T. (1995.) Error-Correcting Output Coding Corrects Bias and Variance In *Machine Learning: Proceedings of the Twelfth International Conference on Machine Learning* Tahoe City, CA: Morgan Kaufmann.

Kononenko, I., & Kovacic, M. (1992.) Learning as Optimization: Stochastic Generation of Multiple Knowledge In *Machine Learning: Proceedings of the Ninth International Workshop* Aberdeen, Scotland: Morgan Kaufmann.

Kovacic, M (1994.) MILP - a stochastic approach to Inductive Logic Programming In *Proceedings of the Fourth International Workshop on Inductive Logic Programming* Bad Honnef/Bonn, Germany: GMD Press.

Kruskal W.H. and Tanur J.M (1978.) *International encyclopedia of statistics.* New York, NY: Free Press.

Kwok, S., & Carter, C. (1990.) Multiple decision trees *Uncertainty in Artificial Intelligence, 4,* 327-335.

Lavrac, N. & Dzeroski, S. (1991.) Inductive learning of relational descriptions from noisy examples In *Proceedings of International Workshop on Inductive Logic Programming ILP-91* Viana de Castelo, Portugal: .

Lloyd J.W. (1984.) *Foundations of Logic Programming.* : Springer-Verlag.

Madigan, D., & York, J. (1993.) *Bayesian Graphical Models for Discrete Data.* (Technical Report UW-93-259). University of Washington, Statistics Department.

Michalski, R.S., & Stepp, R. (1983.) Learning from Observation: Conceptual Clustering. In Michalski, R.S., Carbonell, J.G., & Mitchell T.M. (Ed.s), *Machine Learning: An Artificial Intelligence Approach.* : Tioga Publishing Co.

Muggleton, S., Bain, M., Hayes-Michie, J., & Michie, D. (1989.) An experimental comparison of human and machine-learning formalisms In *Proceedings of the Sixth International Workshop on Machine Learning* Ithaca, NY: Morgan Kaufmann.

Muggleton, S. & Feng, C. (1990.) Efficient Induction of Logic Programs In *Proceedings of the Workshop on Algorithmic Learning Theory* : Japanese Society for Artificial Intelligence.

Muggleton, S., Srinivasan, A., & Bain, M. (1992.) Compression, Significance and Accuracy In *Machine Learning: Proceedings of the Ninth International Workshop* Aberdeen, Scotland: Morgan Kaufmann.

Murphy, P.M., & Aha D.W. (1992.) UCI repository of machine learning databases (a machine-readable data repository). Maintained at the Department of Information and Computer Science, University of Clifornia, Irvine, CA. Data sets are available by anonymous ftp at ics.uci.edu in the directory pub/machine-learning-databases.

Lavrac, N. & Dzeroski, S. (1992.) Background knowledge and declarative bias in inductive concept learning. In Oantke, K., *Proceedings of the Third International Workshop on Analogical and Inductive Inference.* Berlin, Germany: Springer.

Pazzani, M., & Brunk, C. (1991.) Detecting and correcting errors in rule-based expert systems: an integration of empirical and explanation-based learning *Knowledge Acquisition, 3,* 157-173.

Pazzani, M., Brunk, C., & Silverstein, G. (1991.) A knowledge-intensive approach to learning relational concepts In *Machine Learning: Proceedings of the Eighth International Workshop (ML91)* Ithaca, NY: Morgan Kaufmann.

Perrone, M. (1993.) *Improving Regression Estimation: Averaging Methods for Variance Reduction with Extensions to General Convex Measure Optimization.* Doctoral dissertation. Department of Physics, Brown University, .

Quinlan, R. (1986.) Induction of Decision Trees *Machine Learning, 1, 1,* 81-106.

Quinlan, R. (1990.) Learning logical definitions from relations *Machine Learning, 5, 3,* 239-266.

Quinlan, R. (1991.) Technical note: Improved Estimates for the Accuracy of Small Disjuncts *Machine Learning, 6, 1,* 93-98.

Ripley, B.D. (1987.) *Stochastic Simulation.* : John Wiley & Sons.

Schapire, R. (1990.) The strength of Weak Learnability *Machine Learning, 5, 2,* 197-227.

Smyth, P., Goodman, R.M., & Higgins, C. (1990.) A Hybrid Rule-Based/Bayesian Classifier In *Proceedings of the 1990 European Conference on Artificial Intelligence* London, UK: Pitman.

Smyth, P. & Goodman, R. (1992.) Rule Induction Using Information Theory. In G. Piatetsky-Shapiro (ed.), *Knowledge Discovery in Databases.* Menlo Park, CA: AAAI Press, MIT Press.

Spackman, K. (1988.) Learning Categorical Decision Criteria in Biomedical Domains In *Proceedings of the Fifth International Conference on Machine Learning* Ann Arbor, MI: Morgan Kaufmann.

Torgo, L. (1993.) Rule Combination in Inductive Learning In *Machine Learning: ECML 93* Vienna, Austria: Springer-Verlag.

Towell, G., Shavlik, J., & Noordewier, M. (1990.) Refinement of Approximate Domain Theories by Knowledge-Based Artificial Neural Networks In *Proceedings of the Eighth National Conference on Artificial Intelligence (AAAI-90)* Boston, MA: AAAI Press.