A Simpler Look at Consistency †

William M. Spears
Diana F. Gordon
Naval Research Laboratory
Washington, D.C. 20375 USA
(202) 767-9006
spears@aic.nrl.navy.mil
gordon@aic.nrl.navy.mil

Abstract

One of the major goals of most early concept learners was to find hypotheses that were perfectly consistent with the training data. It was believed that this goal would indirectly achieve a high degree of predictive accuracy on a set of test data. Later research has partially disproved this belief. However, the issue of consistency has not yet been resolved completely.

We examine the issue of consistency from a new perspective. To avoid overfitting the training data, a considerable number of current systems have sacrificed the goal of learning hypotheses that are perfectly consistent with the training instances by setting a goal of hypothesis simplicity (Occam's razor). Instead of using simplicity as a goal, we have developed a novel approach that addresses consistency directly. In other words, our concept learner has the explicit goal of selecting the most appropriate degree of consistency with the training data.

We begin this paper by exploring concept learning with less than perfect consistency. Next, we describe a system that can adapt its degree of consistency in response to feedback about predictive accuracy on test data. Finally, we present the results of initial experiments that begin to address the question of how tightly hypotheses should fit the training data for different problems.

Keywords. Consistency bias, simplicity bias, supervised concept learning

---

† Submitted as a technical note to the Special Issue of Machine Learning Journal on "Bias Evaluation and Selection"

1. Introduction

Early studies in supervised concept learning made the implicit assumption that the best method for obtaining high predictive accuracy on a test set is to find hypotheses that are perfectly consistent with respect to all examples in a training set (e.g., Michalski, 1983). A positive hypothesis (i.e., a hypothesis intended to cover the positive examples) is 100% consistent with respect to a set of examples if it covers all positive examples and no negative examples in the set.

Perfect consistency was a goal for many years - until researchers began to examine more realistic databases that contained noisy, sparse data and unknown but possibly complex target concepts. To perform well on these databases, some systems sacrificed perfect consistency in favor of simplicity or other biases (Quinlan, 1987; Michalski, 1990). This achieved excellent results. Today, the issue of the ideal degree of consistency to use in a given situation (e.g., target concept and data characteristics) is still unsettled. Some researchers, such as Angluin and Laird (1988) and Schaffer (1993), discuss the virtues of striving for 100% consistency. Other researchers, such as Quinlan (1987) and Michalski (1990), discuss the virtues of striving for simplicity and thus sacrificing perfect consistency.

A simplicity bias typically satisfies two goals of the person who implements it: improved human understandability of the hypotheses, and improved predictive accuracy by avoiding overfitting the training data. Here, we adopt a novel approach that focuses *only* on the latter goal - we abandon the simplicity bias in favor of a bias that selects a consistency level that yields good predictive accuracy. Each consistency level corresponds to a degree of fit to the training data, where 100% consistency implies the hypothesis fits the training data perfectly.

There are at least three practical reasons for examining levels of consistency instead of simplicity. First, simplicity is not always a straightforward bias to implement. For methods that lean toward biologically motivated learning (e.g., neural networks, genetic algorithms) it may be easier to implement a preference for lower consistency than a preference for simplicity. For example, after choosing a topology for a neural network, it may be much easier to choose a specified level of error after training than to alter the network topology. Likewise, with our genetic algorithm concept learner (described in Section 2), we have found it to be very straightforward to implement a preference for a given consistency level. On the other hand, a method to implement a simplicity preference in this system is still an open and non-trivial question.

Another practical reason for examining a preference for hypotheses that are less than 100% consistent with the training data, rather than a preference for simpler hypotheses, is that the former is a weaker bias. In other words, simplicity usually implies less than 100% consistency, but the converse is not at all true. Thus, although the simplicity bias is a stronger bias and therefore provides a computationally more efficient mechanism for exploring the space of less than 100% consistent hypotheses, it does so at the risk of avoiding many hypotheses that may be potentially useful. This risk becomes larger as concept learning problems become more complex.[1] If one is interested in a domain in which the comprehensibility of hypotheses is *not* an issue, but predictive accuracy is critical, then by weakening the bias it may be possible to find hypotheses that are better predictors than might be found in the (more restricted) set of simple hypotheses.

---

[1] Analogously, greedy search algorithms perform extremely well, until the search space becomes too complicated.

The third justification for examining the consistency bias arises from the fact that different biases may have different regions of expertise in the space of problem/domain characteristics. If we are going to consider a new bias, that of selecting an appropriate consistency level, then we need to perform experiments to identify the regions of expertise of this bias. It is conceivable that in our experimental explorations we will discover that the regions of expertise of this bias differ from those of the simplicity bias. If so, users of machine learning systems will have a reason for selecting one or the other of these two biases, depending on the situation (e.g., the data and other domain characteristics). Here, we focus on initial experiments for characterizing those situations in which lower consistency levels improve predictive accuracy. We also describe a mechanism for dynamically adapting the consistency level based on feedback about predictive accuracy.

In this paper, we examine one learning algorithm and consider the effects of varying the consistency level. Section 2 describes our concept learner, called the Genetic Algorithm Batch-Incremental Learner (GABIL), that we use in all experiments (De Jong et al. 1993). Section 3 describes a modified version of GABIL that can learn concepts with different levels of consistency. This section also presents experimental results that compare predictive accuracy with different consistency levels on both clean and noisy data and a variety of target concepts. Section 4 describes the closed-loop adaptive version of GABIL (AGABIL) that dynamically adjusts its consistency level to improve its predictions. Section 4 presents experimental results on the same data as Section 3, but this time using AGABIL. Section 5 relates this work to other research, and Section 6 states our conclusions and ideas for future research.

2. Background: The GABIL System

In De Jong et al. (1993) we describe GABIL in detail, providing a general architecture for concept learning based on genetic algorithms. In the previous paper we illustrate how some traditional concept learning operators are easily added to the basic GABIL framework, and that GABIL can adapt the use of those operators, using the same underlying genetic mechanism. The result is a simple, yet robust, concept learner. For our experiments regarding the consistency level we have again chosen the GABIL system, this time because it is a very good example of a system in which it is extremely easy to vary the level of consistency of hypotheses but difficult to implement a preference for simplicity. In GABIL, Disjunctive Normal Form (DNF) hypotheses in the form of classification rules compete for survival and reproduce according to their fitness with respect to a set of classified training instances (examples). Those hypotheses that have higher fitness survive and mate, producing new hypotheses via the application of genetic operators.

In GABIL, the concept of fitness is tied to that of consistency as follows:

$$fitness(hyp) = [\ training_accuracy\ (hyp)\ ]^2$$

Accuracy refers to how well a hypothesis predicts the classification of a set of training examples. If a hypothesis predicts all the examples correctly, it is 100% accurate. Similarly, if a hypothesis predicts one half of the examples correctly, it is 50% accurate. Training accuracy is equivalent to consistency. If a hypothesis is *N*% accurate, then it is *N*% consistent. For the sake of understandability, we write the fitness function:

$$fitness(hyp) = [\ consistency\ (hyp)\ ]^2$$

to remind us that we reward those hypotheses that are more consistent. Note that simplicity

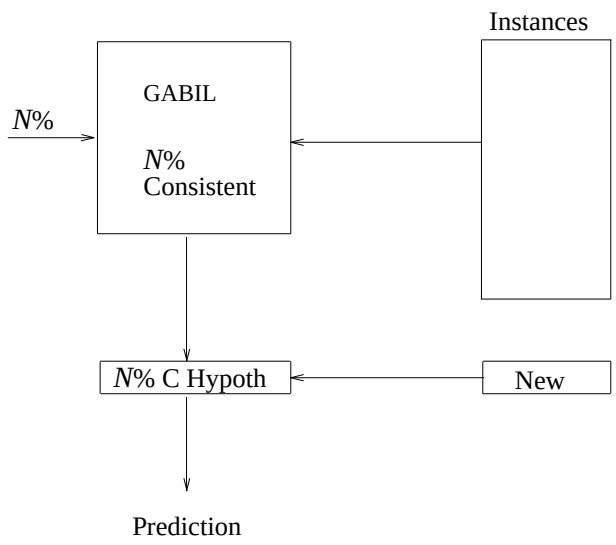

Figure 1. The flowchart for GABIL

plays no role in this fitness function.

A flowchart of GABIL is presented in Figure 1 (in all figures, ‘‘C’’ refers to consistency). GABIL is presented with two inputs: a set of training examples and a desired consistency level of 100% (N = 100 in Figure 1). GABIL returns as output a perfectly consistent hypothesis. This hypothesis is used to predict the classification of a new, previously unseen, example.

3. Varying the consistency level

In this section, we examine the relationship between consistency level and predictive accuracy. To do this, we first modify GABIL so that the system can deliberately select hypotheses with less than 100% consistency. We then test the effects of varying the consistency level.

3.1. Modifications to GABIL

The initial version of GABIL always rewards those hypotheses that are most consistent. Suppose, however, that we wish to consider the effects of lower consistency levels. In other words, suppose we desire a consistency level of 90%. Then we wish to reward most highly those hypotheses that are closest to 90% in accuracy, and to reward less those hypotheses that are both more or less accurate. To do this, we modify the fitness function to:

$$fitness(hyp) = \left[1 - \left| \; consistency(hyp) - N \; \right| \right]^2$$

in which N is the desired consistency level (see Figure 1). This function is maximized when the accuracy (consistency level) of the hypothesis matches N. The fitness is lower for those hypotheses that are both higher and lower than N in accuracy.

3.2. Experimental methodology

We can now use GABIL to compare the effects of consistency level on predictive accuracy. Our experiments use a domain of artificial target concepts, which we call the *n*D*m*C domain. In this domain, we have a four feature world, with four nominal values per feature (i.e., there are 256 instances in this domain). There are eight target concepts, that vary in complexity by increasing both the number of disjuncts and the number of relevant features (conjuncts) per disjunct. The number of disjuncts range from one to four, while the number of conjuncts is either one or three. Each target concept is labeled as *n*D*m*C, where *n* is the number of disjuncts and *m* is the number of conjuncts (see the Appendix for the definition of these target concepts). GABIL learns one target concept at a time. We use an artificial domain because our experimental goal is to find a three-way relationship between the con-

sistency level, target-concept/data characteristics, and predictive accuracy. This is most easily accomplished by methodically varying the target concept and data characteristics on an artificial domain.

Although GABIL is illustrated in Figure 1 as performing in batch mode, it is also capable of performing in a batch-incremental mode (i.e., batch mode is repeated for every new example predicted incorrectly). For each target concept one run consists of running GABIL in batch-incremental mode, over the (randomly shuffled) 256 instances. Each instance is presented once to GABIL. This allows us to generate incremental learning curves for the 256 instances in the *n*D*m*C domain. Each point in an incremental learning curve represents the predictive accuracy over the last 10 instances. Figures 3 - 6 depict a few representative learning curves, where each learning curve is averaged over 10 independent runs. Finally, for the sake of brevity, in our tables we present the global average of the predictive accuracy over each curve. The global average is equivalent to considering the area under the learning curve, and thus represents the performance of GABIL over all the instances of a target concept.

Since the issue of appropriate levels of consistency is intimately tied to that of noise, we examine both noise-free data and data with classification noise. This paper does not examine attribute noise, so we will refer to classification noise as simply ‘‘noise’’. We present results for noise-free data and data with 20% noise. We define *n*% noise such that each instance has a *n*% probability of receiving a random classification (positive or negative). Thus, 100% noise refers to the situation where the target concept is totally obscured. It is important to note that, because every instance is unique, an increase in noise is equivalent to increasing the target concept complexity, and a perfectly consistent hypothesis

is always possible. In this paper, we assume the source is noise. However, we also consider what our experimental results would imply if the source were instead increased target concept complexity.

3.3. Results

As mentioned earlier, our motivation in this section is to examine the effects of consistency level on predictive accuracy. We ran GABIL with consistency levels of 100%, 90%, and 80% on the *n*D*m*C target concepts. Tables 1 - 2 present the global averages of predictive accuracy for each target concept with 0% and 20% noise. We use *N*% to denote GABIL's consistency level in the tables. A ''*'' highlights the winner (i.e., highest predictive accuracy) for each target concept. ''Δ'' is the difference in predictive accuracy between GABIL with 100% and 90% consistency. ''Sig'' is a two-tailed statistical test of significance for that difference.[2]

Table 1. Effect of consistency level

	0% Noise				
TC	100%	90%	80%	Δ	Sig
1D1C	95.4*	85.3	76.5	+10.1	95%
1D3C	96.5*	88.1	79.0	+8.4	95%
2D1C	92.6*	81.1	71.7	+11.5	95%
2D3C	94.1*	88.5	81.5	+5.6	95%
3D1C	90.2*	77.0	69.2	+13.2	95%
3D3C	91.0*	87.8	79.5	+3.2	95%
4D1C	88.8*	75.2	65.8	+13.6	95%
4D3C	88.4*	86.6	79.3	+1.8	90%

[2] The significance test assumes nearly equal variances. We increased the rigor of the significance test whenever the sample variances differed according to the *F*-statistic.

Table 2. Effect of consistency level

	20% Noise				
TC	100%	90%	80%	Δ	Sig
1D1C	78.3	81.5*	74.8	-3.2	95%
1D3C	76.5	83.9*	78.7	-7.4	95%
2D1C	77.2	77.8*	69.9	-0.6	<80%
2D3C	75.8	82.8*	78.4	-7.0	95%
3D1C	77.6*	74.5	67.5	+3.1	90%
3D3C	75.1	80.3*	77.4	-5.2	95%
4D1C	77.2*	74.3	63.0	+2.9	95%
4D3C	73.8	78.9*	77.2	-5.1	90%

Table 1 illustrates a virtue of 100% consistency. With 0% noise, GABIL with 100% consistency performs better than 90% and 80% consistency levels, on all target concepts. With 20% noise, however, the results are reversed. GABIL with 90% consistency outperforms 100% consistency on six of eight target concepts. Clearly, 100% consistency is a disadvantage in this situation.

In summary, as expected, it is certainly not the case that a particular consistency level is most appropriate for all target concepts and amounts of noise. Perfect consistency appears to be appropriate for some situations, and less than perfect consistency is appropriate for others. Therefore, it is natural to ask whether an adaptive mechanism can successfully determine an appropriate level of consistency. We address this issue in the following section.

4. Adaptive consistency level

Recall from Section 3 that GABIL can search for a hypothesis with a desired degree of consistency. In our previous section, this was manually controlled to examine the effect of consistency level on predictive accuracy. Suppose, however, that we could automatically

determine an optimal level, *while* GABIL is learning a particular target concept. The advantages of such a mechanism are two-fold. First, we can analyze the adaptive mechanism to see what level of consistency it chooses for a particular target concept and level of noise. Second, this approach follows the valuable control theory philosophy for closed-loop feedback systems, i.e., if you wish to optimize predictive accuracy, this information should be available to the system. Practically speaking, the resulting system can be more robust because it can monitor its own performance.

4.1. Modifications to GABIL

We modified GABIL to create an adaptive system (AGABIL). AGABIL makes two passes over the training data before predicting the class of each new instance.[3] On the first

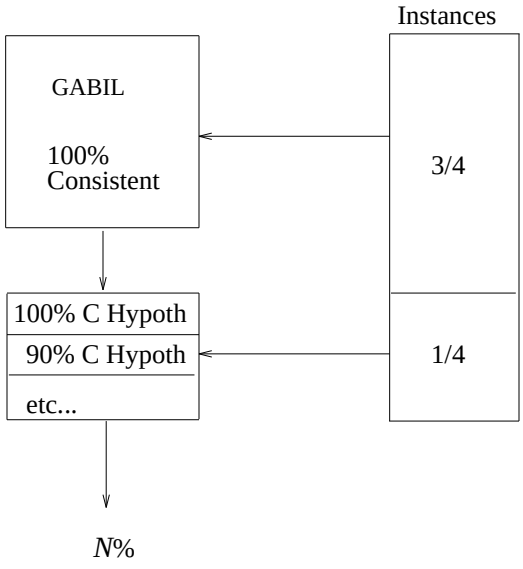

Figure 2. The first pass of AGABIL

[3] Our two-pass method is similar to the cross-validation method described in Breiman et al. (1984). CPU time prohibited the use of more than two passes in AGABIL.

pass, the training data is split into two sets, which we denote as A and B. Set A contains 3/4 of the data, while B contains the remaining 1/4. On the first pass, AGABIL searches for a perfectly consistent hypothesis with respect to set A. AGABIL also stores a small number of less consistent hypotheses as it searches for a 100% consistent hypothesis. In our experiments, AGABIL stores one hypothesis in each of the four ranges: 70-79%, 80-89%, 90-99%, and 100%. AGABIL then compares the predictive accuracy of these stored hypotheses on set B. The consistency level of the best predicting hypothesis is chosen and recorded. The first pass is illustrated in Figure 2.

Next, a second pass over the complete training data is made to find a hypothesis that achieves the chosen consistency level over this data. The hypothesis resulting from the second pass (with the chosen consistency level) is then used to predict the class of a new (test) instance.[4] Figure 1 illustrates the second pass. Note that in this adaptive system the consistency level is no longer set by the user, but is instead determined by a preliminary pass over the training data.

4.2. Results

AGABIL was run on the *n*D*m*C domain, again with 0% and 20% noise. Tables 3 - 4 illustrate the results. In these tables, Δ is the difference in predictive accuracy between AGABIL and GABIL with 100% consistency. Figures 3 - 6 show representative learning curves from which the global predictive accuracy (denoted "PA" in our figures) averages are derived. In these figures, the solid curve is the predictive performance of AGABIL, while the dotted curve is the predictive performance of 100% consistent GABIL.

[4] Experiments in which the resulting first pass hypothesis was used to predict the class of the new (test) instance were not as successful.

Table 3. Performance of adaptive consistency level

	0% Noise				
TC	100%	90%	AGABIL	Δ	Sig
1D1C	95.4	85.3	94.4	-1.0	90%
1D3C	96.5	88.1	96.3	-0.2	<80%
2D1C	92.6	81.1	92.1	-0.5	<80%
2D3C	94.1	88.5	93.9	-0.2	<80%
3D1C	90.2	77.0	89.4	-0.8	<80%
3D3C	91.0	87.8	92.7	+1.7	90%
4D1C	88.8	75.2	88.1	-0.7	<80%
4D3C	88.4	86.6	90.5	+2.1	95%

Table 4. Performance of adaptive consistency level

	20% Noise				
TC	100%	90%	AGABIL	Δ	Sig
1D1C	78.3	81.5	81.9	+3.6	95%
1D3C	76.5	83.9	83.4	+6.9	95%
2D1C	77.2	77.8	79.5	+2.3	90%
2D3C	75.8	82.8	81.7	+5.9	95%
3D1C	77.6	74.5	76.8	-0.8	<80%
3D3C	75.1	80.3	81.0	+5.9	95%
4D1C	77.2	74.3	76.7	-0.5	<80%
4D3C	73.8	78.9	79.2	+5.4	95%

When there is no noise, AGABIL performs well, nearly matching the performance of the best consistency level (100%) on the simpler concepts, and outperforming that consistency level on two of the more difficult concepts. When there is 20% noise, the results indicate quite strongly that the adaptive system can outperform GABIL with 100% consistency. In general, according to Tables 3 - 4, AGABIL performs slightly better in relation to 100% consistent GABIL as the number of conjuncts increases for a fixed number of disjuncts in the tar-

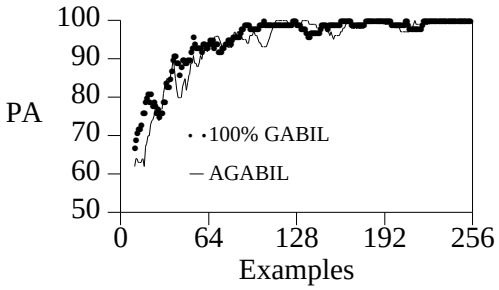

Figure 3. 1D1C - 0% noise

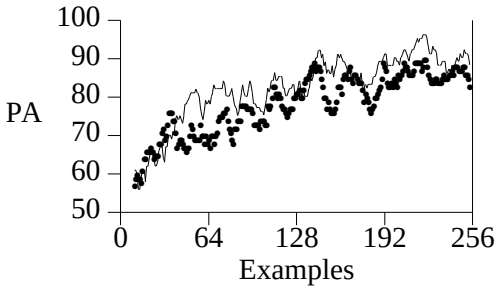

Figure 4. 1D1C - 20% noise

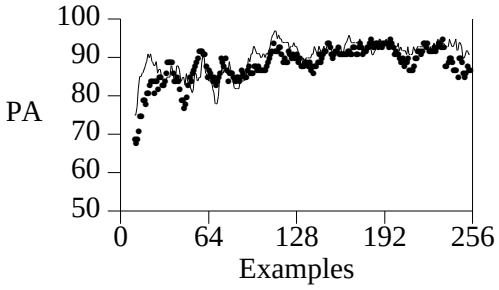

Figure 5. 4D3C - 0% noise

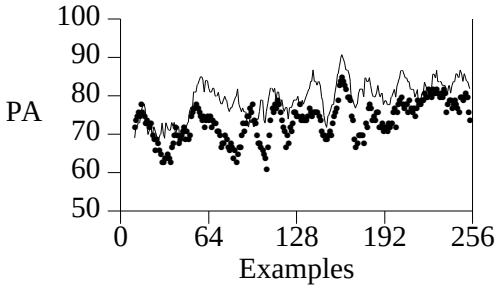

Figure 6. 4D3C - 20% noise

get concept. Tables 3 - 4 also show that when the adaptive system wins, the results tend to have a higher level of statistical significance than when it loses. Furthermore, AGABIL performs competitively with the best consistency level.

These results seem to indicate a trend. AGABIL shows better performance when the noise increases. Recall that we can equate an increase in noise with an increase in target concept complexity. Therefore, we conclude that the adaptive system performs better in relation to the 100% consistent system as noise or target complexity increases.

4.3. A possible caveat

The results in Section 4.2 indicate that lowering the consistency level below 100% is better when the target concept complexity increases. However, these results are particular to

GABIL, which learns only the target concept, and not the negation of the target concept. One could argue that as the target concept becomes more complex, the negation of the target concept may become simpler. In these cases it might be reasonable to learn the negation of the target concept and aim for a consistency level of 100%. If this line of reasoning is correct, we would be entertaining the possibility that the important issue is *not* whether to strive for less than 100% consistency, but rather whether to learn the target concept or the negation of the target concept.

Since the complexity of an arbitrary target concept is not known beforehand, we need some measure to help us determine that complexity as the system runs. One possible measure is the ratio of positive to negative instances. As Schaffer (1993) indicates, the average target concept is simpler for those concepts with a preponderance of positive or negative examples. We examined this ratio for our *n*D*m*C domain and found that only 3D1C and 4D1C have more positive than negative examples. It is interesting to note that, according to the results in Section 4.2, 100% consistency produces better results than lower consistency levels on both 3D1C and 4D1C, regardless of the noise level. This result is true for these target concepts *only*. Therefore, the following heuristic appears to be a valid alternative to selecting the best consistency level:

IF $pos > neg$

THEN learn TC with 100% consistency,

ELSE learn $\neg TC$ with 100% consistency

where pos is the number of positive examples, neg is the number of negative examples, TC is the target concept, and $\neg TC$ is the negation of the target concept. An underlying assumption

of this heuristic is that learning the target concept should produce better results than learning the negation of the target concept when there are more positive examples, and vice versa when there are more negative examples. Another underlying assumption is that 100% *is* the best consistency level for which a system should strive.

Although we have not had time to implement this heuristic within GABIL, we were able to compare the performance of GABIL with consistency levels of 100% and 90%, while learning the negation of the 3D1C and 4D1C target concepts.[5]

The results, which are shown in Table 5, are quite unexpected. First, learning the negation of the target concept is always better than learning the target concept, despite the preponderance of positive examples. Second, there is still evidence that a lower (90%) consistency level is more useful than 100% consistency as the noise increases and the target concept becomes more complex. These results clearly diminish the general usefulness of our heuristic. Therefore, we continue to stress the importance of adaptively adjusting the consistency level.

These results also suggest that it may be difficult to decide *a priori* which target concept for GABIL to learn. One possible solution is to let the system learn both the target

Table 5. Learning the negation

	0% Noise		20% Noise	
TC	100%	90%	100%	90%
3D1C	90.2	77.0	77.6	74.5
¬3D1C	95.7*	85.0	77.6	80.8*
4D1C	88.8	75.2	77.2	74.3
¬4D1C	92.2*	84.0	78.4	81.6*

[5] The negation of 3D1C is a 1D3C concept, and the negation of 4D1C is a 1D4C concept, for this particular domain.

concept and its negation simultaneously. However, there are a number of implementation issues that make this solution infeasible. A more intriguing solution is to let GABIL adaptively decide both its consistency level and the target concept it will learn, based on predictive performance. We will pursue this possibility in future implementations of GABIL.

4.4. Changes in the consistency level

One of the advantages of an adaptive mechanism is robustness. This advantage has been shown in Section 4.2. Another advantage is that the adaptive system can be monitored. Figures 7 - 10 illustrate how the consistency level (denoted "*N*%") within AGABIL changes, for particular target concepts and levels of noise. These figures highlight some interesting points. First, the appropriate level of consistency is lower both for more complex target concepts and for greater levels of noise (which is analogous to a more complex target concept). Second, the consistency level varies more when fewer examples are presented. This indicates that the adaptive mechanism is having some trouble early on, possibly due to insufficient sampling. Finally, and perhaps most interestingly, the consistency level usually drops as the number of examples increases. This behavior appears similar to that described by Fisher and Schlimmer (1988).

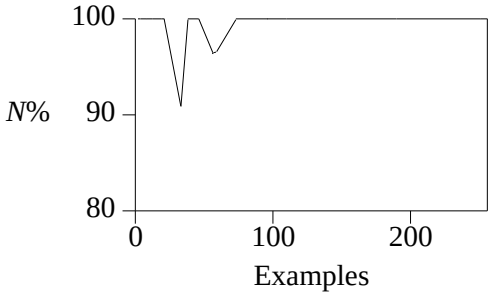

Figure 7. 1D1C - 0% noise

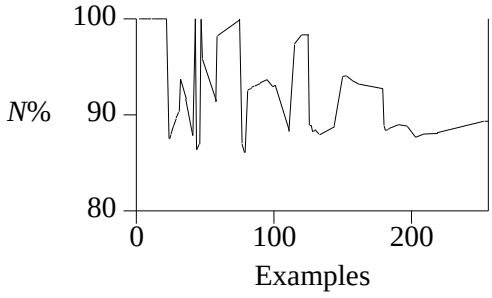

Figure 8. 1D1C - 20% noise

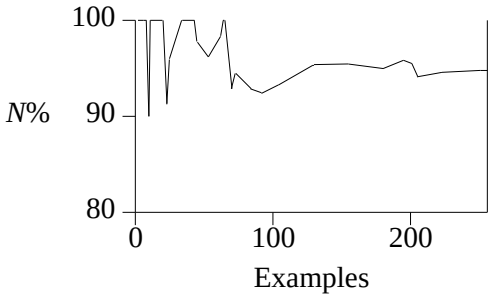

Figure 9. 4D3C - 0% noise

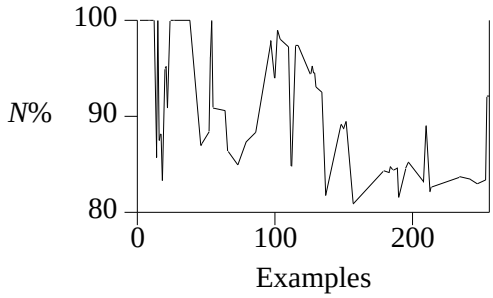

Figure 10. 4D3C - 20% noise

5. Related work

There have been many methods for handling noisy data, such as weighted hypotheses (Schlimmer & Granger, 1986), Bayesian approaches (Buntine, 1991), multiple version spaces (Mitchell, 1978), and tree pruning (Quinlan, 1987; Breiman et al., 1984). The goal of our research is to vary the consistency level to improve predictive accuracy. No previous research has had precisely the same goal.

The most closely related research investigates the effectiveness of a simplicity bias. This research is related because increased simplicity can result in a reduced consistency level. There exists a variety of criteria, such as minimum description-length and reduced entropy, that can be used to achieve hypothesis simplicity (Quinlan & Rivest, 1989; Quinlan, 1986). Simplicity biases have been implemented with two of the most widely used hypothesis representations: decision trees and classification rules. Pruning (e.g., Quinlan, 1987) is a popular method for increasing the simplicity of decision trees. Pruning can reduce the consistency level because each decision tree branch that is pruned away may contain information for distinguishing the classifications of the training instances. After pruning, this information is lost. Breiman et al. (1984), Quinlan (1987), and Fayyad (1993) have demonstrated that pruning is highly effective on some important real-world domains, including

domains containing noisy data. CART, a system developed by Breiman et al. (1984), dynamically varies the amount of pruning by using cross-validation. In this respect CART is similar to AGABIL; however CART differs from AGABIL because it varies simplicity rather than consistency.[6]

The removal of hypothesis disjuncts (e.g., Michalski, 1990) is an effective method to increase the simplicity of DNF hypotheses that are in the form of classification rules. This method may sacrifice 100% consistency because the removed disjuncts may uniquely cover some of the training examples.

As Schaffer (1993) points out, a preference for simplicity is a bias and will therefore be appropriate only in certain situations. Schaffer's goal is to identify situations (e.g., noisiness of the data, complexity of the target concept) favorable for a simplicity bias. Reduced consistency is also a bias, and our experimental results provide an indication of situations in which this bias seems most appropriate. Some of our results, however, seem to conflict with those of Schaffer. For example, AGABIL usually performs better in relation to 100% consistent GABIL as target complexity increases. Also AGABIL usually performs better as classification noise increases. In Schaffer's experiments, CART outperformed 100% consistency when the target concept was simple and there was little classification noise. Further experimentation will be required to determine whether our results conflict because AGABIL's goal and CART's goal differ, or because other biases differ.

A simplicity bias can achieve two learning goals simultaneously: improved predictive accuracy and improved readability. For the sake of experimental clarity, and for the practical

[6] More formally, let I represent a set of instances and H represent the statement that a hypothesis will correctly predict the class of a random instance. We are trying to find a hypothesis that maximizes the a posteriori probability $P(H \mid I)$. Both CART and AGABIL use cross-validation to estimate $P(H \mid I)$, but CART focuses on simplicity to generate hypotheses and AGABIL focuses on consistency.

reasons enumerated in our introduction, we have teased these two goals apart and addressed only the first one. Our results indicate that it is possible to improve predictive accuracy as classification noise or target concept complexity increases by reducing the level of consistency with the training data.

6. Conclusions and future work

In this paper, we have addressed the issue of finding an appropriate consistency level for improving predictive accuracy. Given a suite of target concepts that incrementally increase in complexity, and a corresponding set of training examples that vary in their level of noise, we identify the ‘‘best’’ consistency level for each case. The ‘‘best’’ consistency level is one that yields the highest predictive accuracy. We also describe a method for feeding the predictive accuracy information back into a learner to dynamically adjust the consistency level bias. Finally, we compare the performance of this adaptive system with a system that maintains 100% consistency over the training examples.

From these experiments, we have formed the following conclusions. First, lowering the consistency level seems to be more appropriate as the noise increases. Second, lowering the consistency level also seems to be more appropriate as target concept complexity increases. Finally, we have developed an adaptive concept learner that can select the best consistency level by using predictive accuracy feedback. This adaptive system is novel because it uses the predictive accuracy feedback to select a consistency level, rather than to select a simplicity level.

Future work will focus on four major directions. Our first direction will be to compare the lower consistency bias with the greater simplicity bias, in order to learn when each bias is

more effective. Another direction for future research will be to test the generality of our results by rerunning our experiments using different systems and a wider variety of target concepts. We would like to try alternative hypothesis representations (e.g., decision trees), learning algorithms, and data (e.g., continuous-valued attributes) and target concept characteristics.

A third direction for future research is to determine the effectiveness of an adjustable consistency level for concept learning on real-world domains. To do this, we would like to implement an adjustable consistency level in a sophisticated concept learning system that is well-known for its effectiveness on real-world domains (e.g., the C4.5 system of Quinlan, or CART). Recall that GABIL was chosen for our initial experiments because of the ease with which we could incorporate flexible consistency levels; however, GABIL is not as sophisticated as some of the other concept learning systems. CART would be a good candidate for future experiments because it already has a mechanism to use cross-validation for decision tree selection.

Our fourth direction for future research relates to the results in computational learning theory. Valiant (1984) has introduced the criterion of Probably Approximately Correct (PAC) identification of a target concept. Recently, a number of researchers have considered the computational feasibility of PAC identification in the context of noisy examples (e.g., Angluin & Laird, 1988). However, they assume the strategy is to maximize consistency with the training sample. It would be interesting to also explore the computational feasibility of PAC identification assuming a strategy of lower consistency. We would also like to gain a theoretical understanding of why GABIL improves its predictive accuracy when using lower levels of consistency on more complex target concepts and noisier data. Perhaps an average

case analysis (Pazzani & Sarrett, 1992) would be appropriate for this task.

Finally, we would like to mention that although our experimental results are preliminary, we hope that they will inspire others to further investigate the regions of appropriateness of varied levels of consistency. For some systems, simplicity is a much tougher bias to implement than selecting a consistency level. Furthermore, for some domains, predictive accuracy is essential and readability is not an issue. It will be important to have results about which subsets of those domains are best suited for lower consistency levels, and to have systems that can adaptively select a good consistency level. Also, as mentioned earlier, simplicity is a rather strong bias, and can conceivably prevent a system from exploring other hypotheses that are not perfectly consistent with the training data, are not simple, but are better predictors over future data than simpler hypotheses. As with any search paradigm, it is often of practical importance to weaken biases as problems become more complex. Dropping the readability requirement is one such method of achieving this.

Acknowledgements

Special thanks to Kenneth De Jong, who made many major contributions to the GABIL project. We also thank the members of the Machine Learning Group at NCARAI for their useful comments on this research.

References

Angluin, D. & Laird, P. (1988). Learning from Noisy Examples. *Machine Learning, 2.*

Breiman, L., Friedman, J., Olshen, R., & Stone, C. (1984). *Classification and Regression Trees.* Belmont: Wadsworth.

Buntine, W. (1991). Classifiers: A theoretical and empirical study. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence.*

De Jong, K., Spears, W., & Gordon, D. (1993). Using genetic algorithms for concept learning. *Machine Learning, 14*.

Fayyad, U. (1993). SKICAT: A machine learning system for automated cataloging of large scale sky surveys. In *Proceedings of the Tenth International Conference on Machine Learning.*

Fisher, D. & Schlimmer, D. (1988). Concept Simplification and Prediction Accuracy. In *Proceedings of the Fifth International Conference on Machine Learning*.

Michalski, R. (1983). A theory and methodology of inductive learning. In R. Michalski, J. Carbonell, & T. Mitchell (Eds.), *Machine learning: An Artificial Intelligence Approach* (Vol. 1). Palo Alto: Tioga.

Michalski, R. (1990). Learning flexible concepts: Fundamental ideas and a method based on two-tiered representation. In Y. Kodratoff, R. Michalski (Eds.), *Machine learning: An Artificial Intelligence Approach* (Vol. 3) San Mateo: Morgan Kaufmann.

Mitchell, T. (1978). *Version spaces: An approach to concept learning.* Ph.D. thesis, Stanford University, Stanford, CA.

Pazzani, P. & Sarrett, W. (1992). A framework for average case analysis of conjunctive learning algorithms. *Machine Learning, 9.*

Quinlan, J. (1986). Induction of decision trees. *Machine Learning, 1*.

Quinlan, J. (1987). Simplifying decision trees. *International Journal of Man-Machine Studies, 27.*

Quinlan, J. & Rivest, R. (1989). Inferring decision trees using the minimum description length principle. *Information and Computation, 80.*

Schaffer, C. (1993). Overfitting avoidance as bias. *Machine Learning, 10.*

Schlimmer, J. & Granger, R. (1986). Incremental learning from noisy data. *Machine Learning, 1*.

Valiant, L. (1984). A theory of the learnable. *Communications of the ACM, 27.*

Appendix: *n*D*m*C Target Concepts

This appendix fully describes the target concepts of the artificial domain. There are four features, denoted as F1, F2, F3, and F4. Each feature has four values {v1, v2, v3, v4}.

All the target concepts have the following form:

4D*m*C == d1 v d2 v d3 v d4
3D*m*C == d1 v d2 v d3
2D*m*C == d1 v d2
1D*m*C == d1

For the *n*D3C target concepts we have:

d1 == (F1 = v1) & (F2 = v1) & (F3 = v1)
d2 == (F1 = v2) & (F2 = v2) & (F3 = v2)
d3 == (F1 = v3) & (F2 = v3) & (F3 = v3)
d4 == (F1 = v4) & (F2 = v4) & (F3 = v4)

Finally, we define the *n*D1C target concepts:

d1 == (F1 = v1)
d2 == (F1 = v2)
d3 == (F1 = v3)
d4 == (F1 = v4)