From *Working Notes of the AAAI-94 Workshop on Case-Based Reasoning* (1994). Seattle, WA: AAAI Press.

Oblivious Decision Trees and Abstract Cases

Pat Langley (Langley@flamingo.stanford.edu)
Stephanie Sage (Sage@flamingo.stanford.edu)
Institute for the Study of Learning and Expertise
2451 High Street, Palo Alto, CA 94301

Abstract

In this paper, we address the problem of case-based learning in the presence of irrelevant features. We review previous work on attribute selection and present a new algorithm, Oblivion, that carries out greedy pruning of oblivious decision trees, which effectively store a set of abstract cases in memory. We hypothesize that this approach will efficiently identify relevant features even when they interact, as in parity concepts. We report experimental results on artificial domains that support this hypothesis, and experiments with natural domains that show improvement in some cases but not others. In closing, we discuss the implications of our experiments, consider additional work on irrelevant features, and outline some directions for future research.

1. Introduction

Effective case-based reasoning relies on the identification of a subset of features that are relevant to the learning task. Most work on this topic assumes the developer makes this decision, but application of case-based methods to complex new domains would be aided by automated methods for feature selection. Some researchers (e.g., Barletta & Mark, 1988; Cain, Pazzani, & Silverstein, 1991) have explored the use of domain-specific background knowledge to select useful features, but this approach will not work when little domain knowledge is available. Domain-independent methods for feature selection would augment the techniques available for developing case-based systems.

Rather than selecting features, one might employ all available features during case retrieval, giving them equal weight in this process. Cover and Hart (1967) have proven that a simple nearest neighbor algorithm, probably the simplest case-based method, has excellent asymptotic accuracy. However, more recent theoretical analyses (Langley & Iba, 1993) and experimental studies (Aha, 1990) suggest that the empirical sample complexity of nearest neighbor methods is *exponential* in the number of irrelevant features. This means that the presence of irrelevant attributes can slow the *rate* of case-based learning drastically.

A natural response is to draw on machine learning techniques to identify those attributes relevant to the task at hand. For example, Cardie (1993) used a decision-tree method (C4.5) to select features for use during case retrieval. She passed on to a k nearest neighbor algorithm only the features occurring in the induced decision tree. She reported good results in a natural language domain, with k nearest neighbor in the reduced space outperforming both C4.5 and k nearest neighbor using all the features.

Unfortunately, although the greedy approach of C4.5 works well for conjunctive and m of n concepts, it suffers when attribute interactions exist. In this case, a relevant feature in isolation may appear no more discriminating than an irrelevant one. Parity concepts constitute the most extreme example of this situation. Experimental studies (Almuallim & Dietterich, 1991; Kira & Rendell, 1992) confirm that, for some target concepts, decision-tree methods deal poorly with irrelevant features.

Almuallim and Dietterich's Focus (1990) tried to address this difficulty by searching for combinations of features that discriminate the classes. The accuracy of this method is almost unaffected by the introduction of irrelevant attributes, but its time complexity is quasi-polynomial in the number of attributes. Schlimmer (1993) presented a related technique that uses knowledge about the partial ordering of the space to reduce the search, but still had to limit the complexity of learnable target concepts to keep the search within bounds. Thus, there remains a need for more practical algorithms that can handle domains with complex feature interactions and irrelevant attributes.

In the following pages, we present a new algorithm – Oblivion – that should handle irrelevant features in a more efficient manner than Almuallim and Dietterich's or Schlimmer's techniques, and we show how the method can be viewed as identifying and storing abstract cases. We report experimental studies of Oblivion's behavior on both artificial and natural domains, and we draw some tentative conclusions about the approach to feature selection it embodies. Finally, we consider some additional related work and suggest directions for future research on this topic.

2. Induction of Oblivious Decision Trees

Our research goal was to develop an algorithm that handled both irrelevant features and attribute interactions without resorting to expensive, enumerative search. Our response draws upon the realization that both Almuallim and Dietterich's and Schlimmer's approaches construct *oblivious* decision trees, in which all nodes at the same level test the same attribute. Although these methods use forward selection (i.e., top-down search) to construct oblivious decision trees, one can also start with a full oblivious decision tree that includes all the attributes, and then use pruning or backward elimination to remove features that do not aid classification accuracy. The advantage of the latter approach is that accuracy decreases substantially when one removes a *single* relevant attribute, even if it interacts with other features, but remains unaffected when one prunes an irrelevant or redundant feature.

Oblivion is an algorithm that instantiates this idea. The method begins with a full oblivious tree that incorporates all potentially relevant attributes and estimates this tree's accuracy on the entire training set, using a conservative technique like n-way cross validation. Oblivion then removes each attribute in turn, estimates the accuracy of the resulting tree in each case, and selects the most accurate. If this tree makes no more errors than the initial one, Oblivion replaces the initial tree with it and continues the process. On each step, the algorithm tentatively prunes each of the remaining features, selects the best, and generates a new tree with one fewer attribute. This continues until the accuracy of the best pruned tree is less than the accuracy of the current one. Unlike Focus and Schlimmer's method, Oblivion's time complexity is polynomial in the number of features, growing with the square of this factor.

There remain a few problematic details, such as constructing an initial tree that is exponential in the number of initial attributes, determining the order of the retained attributes, and passing the results to some learning method. However, none of these steps is actually necessary. The key lies in realizing that an oblivious decision tree is *equivalent* to a nearest neighbor scheme that ignores some features. In this view, each path through the tree corresponds to an abstract case that summarizes an entire set of training instances. Because pruning can produce impure partitions of the training set, each such case specifies a distribution of class values. When an instance matches a case's conditions, it simply predicts the most likely class. If training data are sparse and a test instance fails to match any stored abstract case, one finds the nearest cases (i.e., with the most matched conditions), sums the class distributions for each one, and predicts the most likely class. This insight into the relation between oblivious decision trees and nearest neighbor algorithms was an unexpected benefit of our work.

3. Experimental Studies of Oblivion

We expected Oblivion to scale well to domains that involve many irrelevant features. To test this prediction, we designed an experimental study with four artificial Boolean domains that varied both the degree of feature interaction and the number of irrelevant features. We examined two target concepts – five-bit parity and a five-feature conjunction – in the presence of both zero and three irrelevant attributes. For each condition, we randomly generated 20 sets of 200 training cases and 100 test cases, and measured classification accuracy on the latter. In addition to varying the two domain characteristics, we also examined three induction algorithms – simple nearest neighbor (which does not carry out attribute selection), C4.5 (which employs a forward greedy selection), and Oblivion (i.e., nearest neighbor with backward greedy selection). Finally, we varied the number of training instances available before testing, to obtain learning curves.

We had a number of hypotheses about the outcomes of this study. First, we expected C4.5 to be unaffected by irrelevant attributes in the conjunctive domain, but to suffer on the parity concept, because none of the five relevant features would appear diagnostic in isolation. In contrast, we predicted that nearest neighbor would suffer equally on both target concepts, but that Oblivion's ability to remove irrelevant features even in the presence of feature interaction would let it scale well on both concepts. Finally, we hypothesized that Oblivion's learning curve would closely follow that for nearest neighbor when no irrelevants were present, but that it would mimic C4.5 in the absence of feature interactions.

Figure 1 (a) shows the learning curves on the parity target concept when only the five relevant attributes and no irrelevant ones are present in the data. In this experimental condition, nearest neighbor and Oblivion increase their accuracy at the same rate, but surprisingly, C4.5 actually learns somewhat more rapidly. The situation changes drastically in Figure 1 (b), which presents the results when there are three irrelevant features. Here the learning curves for both nearest neighbor and C4.5 have flattened considerably. In contrast, the learning rate for Oblivion is almost unaffected by their introduction. A different situation holds for the conjunctive target concept (not shown). In this case, all three algorithms require about the same number of instances to reach perfect accuracy when no irrelevants are present, with nearest neighbor taking a surprise lead in the early part of training. The introduction of irrelevant attributes affects nearest neighbor the most, and C4.5's learning curve is somewhat less degraded than that for Oblivion.

These results support our hypothesis about Oblivion's ability to scale well to domains that have both irrelevant features and interaction among relevant at-

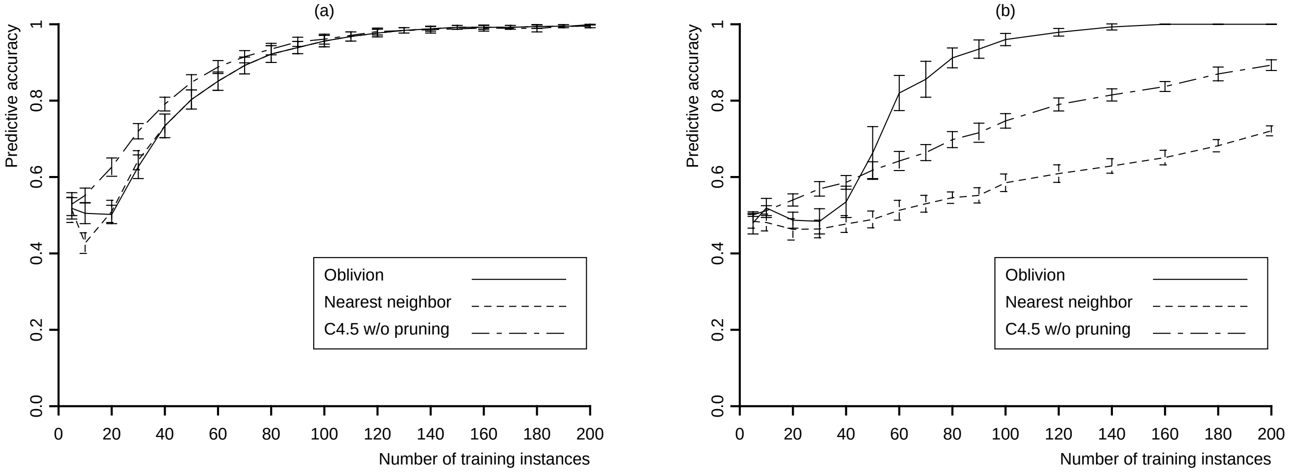

Figure 1. Learning curves for nearest neighbor, C4.5 without pruning, and Oblivion on the five-bit parity concept given (a) zero irrelevant attributes and (b) three irrelevant attributes. The error bars indicate 95% confidence intervals.

tributes. However, we also wanted to evaluate the importance of this finding on natural data. Holte's (1993) results with the UCI repository suggest that these domains contain many irrelevant features but few interactions among relevant ones; in this case, we would expect C4.5 and Oblivion to outperform nearest neighbor on them. But it is equally plausible that these domains contain many relevant but *redundant* attributes, in which case we would observe little difference in learning rate among the three algorithms.

In four of the UCI domains – Congressional voting, mushroom, DNA promoters, and breast cancer – we found little difference in the behavior of Oblivion, C4.5, and nearest-neighbor. All three algorithms learn rapidly and the learning curves (not shown) are very similar. Inspection of the decision trees learned by C4.5 and Oblivion in two of these domains revealed only a few attributes. Combined with the fact that nearest neighbor performs at the same level as the other methods, this is consistent with the latter explanation for Holte's results, that these domains contain largely redundant features.[1]

One domain in which Holte found major differences was king-rook vs. king-pawn chess endgames, a two-class data set that includes 36 nominal attributes. This suggested that it might contain significant attribute interactions, and thus might give different outcomes for the three algorithms. Figure 2 (a) gives the resulting learning curves, averaged over 20 runs, in which Oblivion's accuracy on the test set is consistently about ten percent higher than that for nearest neighbor, though presumably the latter would eventually catch up if given enough instances. However, C4.5 reaches a high level of accuracy even more rapidly than Oblivion, suggesting that this domain contains many irrelevant attributes, but that there is little interaction among the relevant ones. Inspection of the decision trees that C4.5 generates after 500 instances is consistent with this account, as they contain about ten of the 35 attributes, but only a few more terminal nodes than levels in the tree, making them nearly linear and thus in the same difficulty class as conjunctions.

Figure 2 (b) shows encouraging results on another domain, this time averaged over ten runs, that involves prediction of a word's specific semantic class from the surrounding context in the sentence. These data include 35 nominal attributes (some with many possible values) and some 40 word classes. Nearest neighbor does very poorly on this domain, suggesting that many of the attributes are irrelevant. Inspection of C4.5's and Oblivion's output, which typically retain about half of the attributes, is consistent with this explanation. In the latter part of the learning curves, Oblivion's accuracy pulls slightly ahead of that for C4.5, but not enough to suggest significant interaction among the relevant attributes. Indeed, Cardie (1993) reports that (on a larger training set) nearest neighbor outperforms C4.5 on this task when the former uses only those features found in the latter's decision tree. This effect cannot be due to feature interaction, since it relies on C4.5's greedy forward search to identify features; instead, it may come from the different representational biases of decision trees and case-based methods, which would affect behavior on test cases with imperfect matches.

The above findings indicate that many of the available data sets contain few truly irrelevant features, and none of these appear to involve complex feature interactions. These observations may reflect preprocessing

[1] A forward-selection variant of Oblivion (basically a greedy version of the Focus algorithm) also produced very similar curves on these domains, providing further evidence that they do not involve both feature interactions and irrelevant attributes.

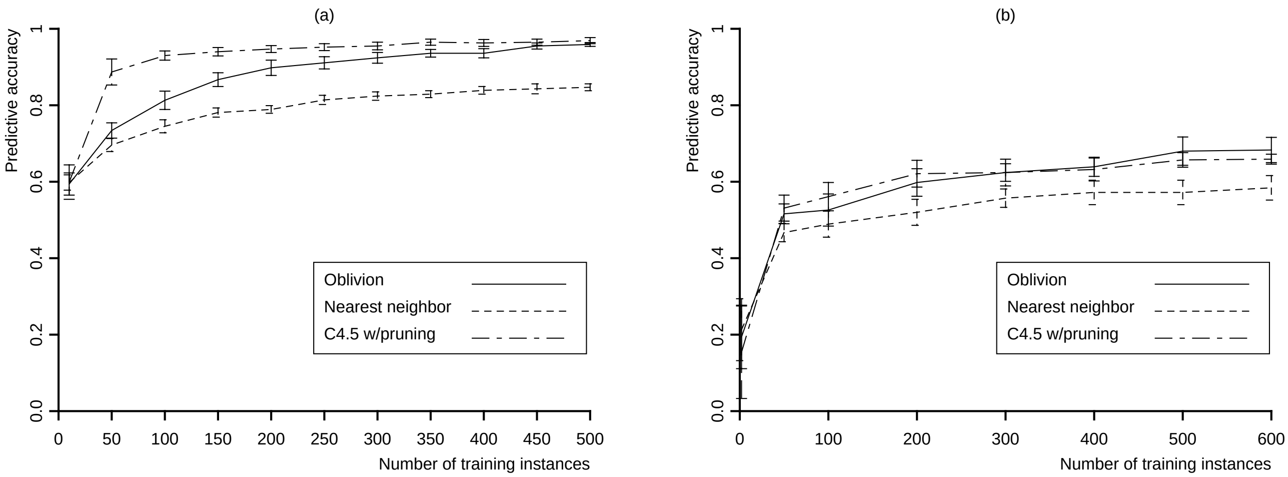

Figure 2. Predictive accuracy as a function of training instances for nearest neighbor, C4.5 with pruning, and Oblivion on (a) classifying chess endgames and (b) predicting a word's semantic class.

of many of the UCI databases by domain experts to remove irrelevant attributes and to replace interacting features with better terms. The voting records, which contain only 16 *key* votes as identified by the *Congressional Quarterly*, provide an extreme example of the first trend. As machine learning starts to encounter new domains in which few experts exist, such data sets may prove less representative than artificial ones.

The experiments with artificial domains, reported earlier, revealed clear differences in the effect of irrelevant attributes and feature interactions on the behavior of nearest neighbor, C4.5, and Oblivion. The rate of learning for the nearest neighbor method decreased greatly with the addition of irrelevant features, regardless of the target concept. In contrast, irrelevant attributes hurt C4.5 for the five-bit parity concept but not the five-feature conjunction; top-down greedy induction of decision trees scales well only when the relevant features (individually) discriminate among the classes. In contrast, the learning rate for Oblivion was largely unaffected by irrelevant features for either the conjunctive or parity concepts, presumably because its greedy pruning method was not misled by interactions among the relevant features.

4. Discussion

We have already reviewed the previous research that led to our work on Oblivion, and we have drawn some tentative conclusions about the algorithm's behavior from our experimental results. Here we consider some additional related work on induction, along with directions for future research.

Kira and Rendell (1992) have followed a somewhat different approach to feature selection. For each attribute A, their Relief algorithm assigns a weight W_A that reflects the relative effectiveness of that attribute in distinguishing the classes. The system then selects as relevant only those attributes with weights that exceed a user-specified threshold, and passes these features, along with the training data, to another induction algorithm such as ID3. Comparative studies on two artificial domains with feature interactions showed that, like Focus, the Relief algorithm was unaffected by the addition of irrelevant features on noise-free data, and that it was less affected than Focus (and much more efficient) on noisy data.

The above algorithms *filter* attributes before passing them to ID3, but John, Kohavi, and Pfleger (in press) have explored a *wrapper* model that embeds a decision-tree algorithm within the feature selection process, and Caruana and Freitag (in press) have described a similar scheme. Each examined greedy search through the attribute space in both the forward and backward directions, including variants that supported bidirectional search. John et al. found that backward elimination produced more accurate trees than C4.5 in two domains but no differences in others, whereas Caruana and Freitag reported that all of their attribute-selection methods produced improvements over (unpruned) ID3 in a single domain.

One can also combine the wrapper idea with nearest-neighbor methods, as in Oblivion. Skalak (in press) has recently examined a similar approach, using both Monte Carlo sampling and random mutation hill climbing to select cases for storage, with accuracy on the training set as his evaluation measure. Both approaches led to reductions in storage costs on four domains and some increases in accuracy, and the use of hill climbing to select features gave further improvements. Moore, Hill, and Johnson (in press) have also embedded nearest neighbor methods within a wrapper scheme. However, their approach to induction searches not only the space of features, but also the number of neighbors used in prediction and the space of combination functions. Using a leave-one-out scheme to estimate accuracy on the test set, they have achieved significant results on two control problems that involve the prediction of numeric values.

Some researchers have extended the nearest neighbor approach to include weights on attributes that modulate their effect on the distance metric. For example, Cain et al. (1991) found that weights derived from a domain theory increased the accuracy of their nearest-neighbor algorithm. Aha (1990) presented an algorithm that learned the weights on attributes, and showed that its empirical sample complexity grew only linearly with the number of irrelevant features, as compared to exponential growth for simple nearest neighbor. In principle, proper attribute weights should produce more accurate classifiers than variants that simply omit features. However, search through the weight space involves more degrees of freedom than Oblivion's search through the attribute space, making their relative accuracy an open question for future work.

Clearly, our experimental results are somewhat mixed and call out for additional research. Future studies should examine other natural domains to determine if feature interactions arise in practice. Also, since Oblivion uses the leave-one-out scheme to estimate accuracy, we predict it should handle noise well, but we should follow Kira and Rendell's lead in testing this hypothesis experimentally. Oblivion's simplicity also suggests that an average-case analysis would prove tractable, letting us compare our experimental results to theoretical ones. We should also compare Oblivion's behavior to other methods for selecting relevant features, such as those mentioned above.

Despite the work that remains, we believe that our analysis has revealed an interesting relation between oblivious decision trees and abstract cases, and that our experiments provide evidence that one such algorithm outperforms simpler case-based learning methods in domains that involve irrelevant attributes. We anticipate that further refinements to Oblivion will produce still better results, and that additional experiments will provide a deeper understanding of the conditions under which such an approach is useful.

Acknowledgements

Thanks to David Aha, George John, Karl Pfleger, Russ Greiner, Ronny Kohavi, Bharat Rao, and Jeff Schlimmer for useful discussions, to Ray Mooney, for making his modified C4.5 code available, and to Claire Cardie, for providing her natural language data. Siemens Corporate Research and Stanford University provided resources that aided our research. This work was supported in part by ONR Grant No. N00014-94-1-0505.

References

Aha, D. (1990). *A study of instance-based algorithms for supervised learning tasks: Mathematical, empirical, and psychological evaluations*. Doctoral dissertation, Department of Information & Computer Science, University of California, Irvine.

Almuallim, H., & Dietterich, T. G. (1991). Learning with many irrelevant features. *Proceedings of the Ninth National Conference on Artificial Intelligence* (pp. 547–552). San Jose, CA: AAAI Press.

Barletta, R., & Mark, W. (1988). Explanation-based indexing of cases. *Proceedings of the Seventh National Conference on Artificial Intelligence* (pp. 541–546). St. Paul, MN: AAAI Press.

Cain, T., Pazzani, M. J., & Silverstein, G. (1991). Using domain knowledge to influence similarity judgements. *Proceedings of the DARPA Workshop on Case-Based Reasoning* (pp. 191–199). Washington, DC: AAAI Press.

Cardie, C. (1993). Using decision trees to improve case-based learning. *Proceedings of the Tenth International Conference on Machine Learning* (pp. 25–32). Amherst, MA: Morgan Kaufmann.

Cover, T. M., & Hart, P. E. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, *13*, 21–27.

Holte, R. (1993). Very simple classification rules perform well on most commonly used domains. *Machine Learning*, *11*, 63–91.

Kira, K., & Rendell, L. (1992). A practical approach to feature selection. *Proceedings of the Ninth International Conference on Machine Learning* (pp. 249–256). Aberdeen, Scotland: Morgan Kaufmann.

Langley, P., & Iba, W. (1993). Average-case analysis of a nearest neighbor algorithm. *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence* (pp. 889–894). Chambery, France.

Moore, A. W., Hill, D. J., & Johnson, P. (in press). An empirical investigation of brute force to choose features, smoothers, and function approximators. In S. Hanson, S. Judd, & T. Petsche (Eds.), *Computational learning theory and natural learning systems* (Vol. 3). Cambridge, MA: MIT Press.

Quinlan, J. R. (1986). *C4.5: Programs for machine learning*. San Mateo, CA: Morgan Kaufmann.

Schlimmer, J. C. (1987). Efficiently inducing determinations: A complete and efficient search algorithm that uses optimal pruning. *Proceedings of the Tenth International Conference on Machine Learning* (pp. 284–290). Amherst, MA: Morgan Kaufmann.