Compositional Instance-Based Acquisition of Preference Predicates

Patrick Broos and **Karl Branting**
Department of Computer Science
University of Wyoming
Laramie, Wyoming 82071-3682
patb@pingora.uwyo.edu karl@eolus.uwyo.edu
(307) 766-4258

Abstract

Knowledge to guide search can be represented as a preference predicate $P_Q(x, y)$ expressing that state x is preferable to state y. Interactions by a learning apprentice with a human expert provide an opportunity to acquire exemplars consisting of pairs that satisfy P_Q. CIBL (compositional instance-based learning) is a strategy for learning preference predicates that permits multiple exemplars to be composed, directly exploiting the transitivity of P_Q. An empirical evaluation with artificial data showed that CIBL is consistently more accurate than an instance-based learning strategy unable to compose exemplars. Moreover, CIBL outperforms decision tree induction when the evaluation function Q underlying P_Q contains one or more extrema or a severe discontinuity.

Introduction

Heuristic search is guided by estimations of the relative desirability of alternative states in a state space. For example, A^* uses an evaluation function $f(n)$ to determine the node n_i for which the minimal cost path constrained to go through n_i is estimated to be least. Knowledge of the relative desirability of two states can be expressed as a *preference predicate* [Utgoff and Saxena, 1987] $P_Q(x, y) \equiv [Q(x) > Q(y)]$, where $Q(n)$ is an evaluation function, like A^*'s $f(n)$, that expresses the "quality" of state n.

Acquisition of a preference predicate P_Q for a given state space obviates the task of acquiring the specific underlying evaluation function Q, because Q is useful in search only for selecting the best alternative state. Moreover, in the context of learning apprentice systems,[1] information about an evaluation function may be acquired in the form of pairs (x, y) for which $P_Q(x, y)$. For example, each time a learning apprentice suggests a state s_1 and the expert rejects s_1 in favor of some other state s_2, the apprentice has an opportunity to acquire the training instance $P_Q(s_2, s_1)$.

Our interest in acquisition of preference predicates arises from a project to develop an intelligent assistant for scheduling ground-based telescope observations, the *Observing Assistant.* A fundamental requirement for designing useful observation schedules is the ability to distinguish which of any pair of alternative schedules is preferable. However, we found that astronomers frequently lack clearly articulated criteria for preferring one schedule over another. Moreover, it appears that individual astronomers may differ in their preferences. These factors suggested that devising an *a priori* evaluation function for observation schedules would be difficult and would lead to inflexibility.

However, interactions with an astronomer using an observing assistant provide an opportunity to acquire training instances: whenever a user rejects the observing assistant's suggested schedule, $schedule_1$ in favor of some alternative, $schedule_2$, the observing assistant can acquire the training instance $P_Q(schedule_2, schedule_1)$. Thus, a central component of the task of learning how to design observation schedules appropriate for a particular astronomer consists of acquiring a preference predicate P_Q from a set of training instances generated during interaction with the astronomer.

Previous approaches to acquisition of preference predicates from sets of training instances have used inductive learning methods to form generalizations from sets of training instances [Utgoff and Saxena, 1987; Utgoff and Clouse, 1991]. One approach has been to use decision tree induction algorithms, such as ID3 [Quinlan, 1986], to induce a general representation for P_Q. An alternative approach, termed the *state preference method*, uses parameter adjustment to learn a set of feature weights $\mathbf{W}$ such that for every training instance, $P_Q(x, y)$, $\mathbf{W}(\mathbf{F}(x) - \mathbf{F}(y)) > 0$, where $\mathbf{F}(n)$ is a vector of numeric attributes representing state n [Utgoff and Clouse, 1991].

However, there is reason to believe that the underly-

[1] Learning apprentice systems have been defined as "interactive knowledge-based consultants that directly assimilate new knowledge by observing and analyzing the problem-solving steps contributed by their users through their normal use of the system" [Mitchell *et al.*, 1985].

ing evaluation function Q for astronomical observation schedules, like preference predicates in many other domains [Callan *et al.*, 1991], is typically not linear, that is, the instances of P_Q are not linearly separable. As a result, learning algorithms that presuppose linear separability, such as the state preference method, are inappropriate for this domain.

Decision tree induction algorithms such as ID3 are suitable for nonlinearly separable data. However, the performance of decision tree induction algorithms is often weaker than that of instance-based algorithms when the training set is sparse or the concept being acquired is highly "polymorphic" [Aha, 1992]. Since we believe that these factors will often characterize acquisition of observation scheduling preference predicates, our attention turned to instance-based approaches to this problem.

Instance-Based Learning of Preference Predicates

Instance-based learning (IBL) is a strategy in which concepts are represented by exemplars rather than by generalizations induced from those exemplars [Aha, 1990; Stanfill and Waltz, 1986]. Perhaps the simplest form of instance-based learning is k-nearest-neighbor (k-NN) classification, which classifies a new instance according to the majority classification of its k nearest neighbors in feature space. In most recent IBL systems, $k = 1$ [Aha, 1990].

We have implemented a straightforward 1-NN strategy for learning preference predicates, which we term *1ARC*. 1ARC forms a model of a preference predicate, given training instances that consist of pairs of objects for which the preference predicate P_Q is satisfied. Each training instance $P_Q(X,Y)$ is represented as a directed arc $Y \rightarrow X$ in feature space. We will adopt the notation $\overleftarrow{XY}$; note that arcs point "uphill" towards objects of higher quality, *i.e.*, the head of an arc is preferred to the tail. For example, on a two dimensional feature space $S = \Re^2$, the set of training instances $\{P_Q(A,B), P_Q(C,D), P_Q(E,F)\}$ is represented by the training arcs $\overleftarrow{AB}$, $\overleftarrow{CD}$, and $\overleftarrow{EF}$ as shown in Figure 1. These training instances represent relationships such as "object A is preferred over object B".

Suppose a new pair of objects, X and Y, need to be ranked. Ranking X and Y is equivalent to determining which of the predicates $P_Q(X,Y)$ or $P_Q(Y,X)$ is satisfied. The 1ARC algorithm begins by finding the training instance (training arc) that best matches the hypothesis $P_Q(X,Y) \equiv \overleftarrow{XY}$. The dissimilarity between the proposed arc $\overleftarrow{XY}$ and a training arc is measured by the sum of the euclidean distances between their two heads and their two tails. For example, as shown in Figure 2, the training arc $\overleftarrow{EF}$ best matches $\overleftarrow{XY}$ with a dissimilarity of $dist(Y,F) + dist(X,E)$ represented by the dotted lines. The dissimilarity of the best match is a measure in the confidence in the hypothesis $P_Q(X,Y)$.

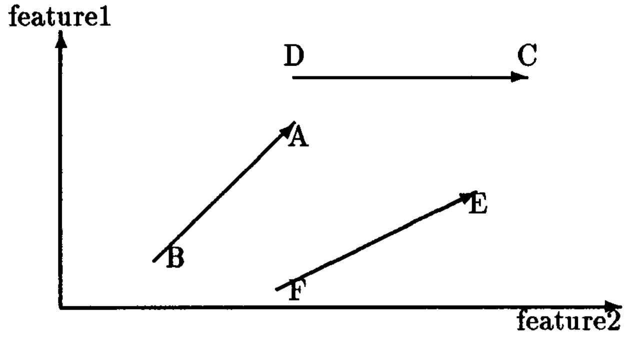

Figure 1: Training arcs.

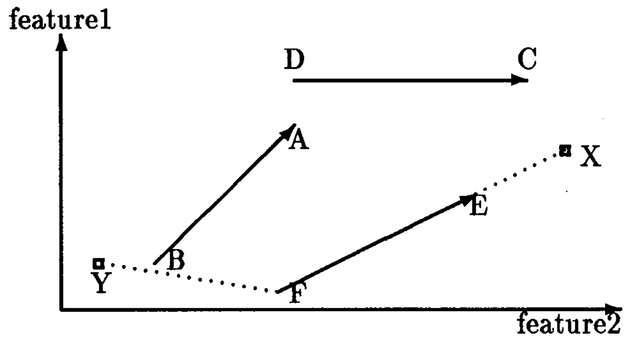

Figure 2: Best match to $\overleftarrow{XY}$.

In a similar fashion 1ARC then finds the best match and confidence measure for the alternate hypothesis $P_Q(Y,X)$. The stronger of the two hypotheses determines 1ARC's estimate of the ranking between X and Y.

A natural extension to 1ARC is to exploit the transitivity of P_Q. For example, given the situation in Figure 2, it should be possible to conclude $P_Q(X,Y)$ by the reasoning "X is close to C; C is preferred to D; D is close to A; A is preferred to B; B is close to Y". This composition of instances is different from the majority vote policy of standard k-NN methods for $k > 1$. Such an extension is described in the next section.

Compositional Instance-Based Learning

CIBL (Compositional Instance-Based Learning) is a strategy that permits multiple training instances to be composed to rank a new pair of objects. Just like 1ARC, CIBL forms a model of a preference predicate, given training instances represented by arcs such as $\overleftarrow{AB}$. When asked to rank two new objects, X and Y, CIBL attempts to find a path from Y to X by composing training arcs, that is, by connecting them sequentially. Such a path seeks to follow a contour of the underlying evaluation function that has positive slope, to establish that object X is preferred to object Y. CIBL also attempts to find a path from X to Y to establish that object Y is preferred over object X. The better of the two paths found determines the ranking of X and Y. This search for paths in feature space that follow arcs can be reduced to a simple search for paths between nodes in a conventional graph.

To illustrate this process, suppose again we have the training arcs shown in Figure 1. Given two new objects to rank, X and Y, CIBL begins by searching for a path from Y to X, supporting the hypothesis $P_Q(X,Y) \equiv \overleftarrow{XY}$. As shown in Figure 3, to construct a path from Y to X, CIBL forms the uncertain arcs G1, G2, and G3. The uncertain arc G1 connects the starting object, Y, with the tail of a nearby training arc, B. The uncertain arc G2 connects the head of a training arc, A, with the tail of another training arc, D. The uncertain arc G3 connects the head of a training arc, C, with the goal object, X. The cost of the resulting path from Y to X is the sum of the euclidean lengths of the uncertain arcs G1, G2, and G3; the training arcs have zero cost. In a similar fashion, a path is constructed from X to Y. The best path, corresponding to the best estimate of the ranking of X and Y, is taken to be the shorter path. In forming a path that uses multiple training arcs, CIBL is composing multiple chunks of knowledge gained from previous experience.

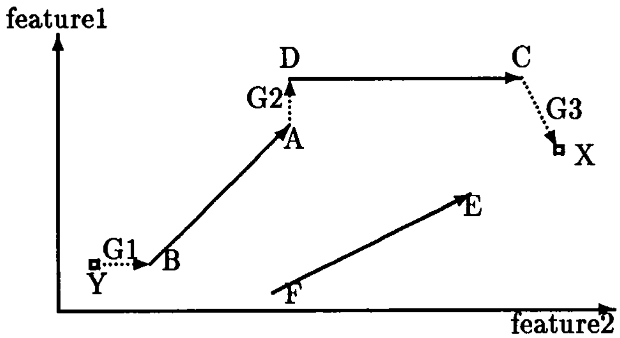

Figure 3: Uncertain arcs added.

CIBL constructs an uncertain arc from the head of each training arc to the tail of every other training arc, forming a dense graph with two special nodes, X and Y. The standard Dijkstra algorithm [Aho *et al.*, 1974] is then used to find the lowest cost path connecting X and Y, where edges representing training arcs are assigned zero cost and edges representing the uncertain arcs are assigned a cost equal to their euclidean length. For clarity, Figure 3 omits some uncertain arcs.

The assumption underlying the formation of an uncertain arc from the head of a training arc to the tail of another training arc is that objects in feature space that are near each other are more likely to have similar quality, as measured by the underlying evaluation function, than objects that are farther apart. The formation of uncertain arcs may be viewed as a kind of local or "lazy" generalization. An uncertain arc may, of course, be erroneous; its head may not be preferred over its tail. The degree to which an uncertain arc may misrank its head and tail depends on the steepness of the underlying evaluation function in that area. If CIBL evaluates the predicate $P_Q(X,Y)$ using a path that contains erroneous uncertain arcs then the truth value of $P_Q(X,Y)$ that CIBL reports may be incorrect.

Empirical Evaluation

We anticipated that CIBL's ability to compose multiple instances would lead to better performance than 1ARC under all conditions except extremely low numbers of training instances, when they would be equivalent. Moreover, we hypothesized that CIBL would outperform decision tree induction for "irregular" evaluation functions, *e.g.,* those having discontinuities or changes in the sign of their derivative.

To test these hypotheses, we compared the accuracy of CIBL to that of 1ARC and ID3 on the task of learning preference functions, P_Q, for a variety of evaluation functions, Q. All but one of the evaluation functions were defined on the feature space $S = [0,1] \times [0,1]$. ID3 was modified to handle real-valued features in the manner discussed by Quinlan [Quinlan, 1986]. For each function, we randomly generated instances of the associated preference predicate of the form $< +, X, Y >$ or $< -, X, Y >$, representing the knowledge "X is preferred over Y" or "X is not preferred over Y" for $X, Y \in S$. Each model was trained on a set of instances of size $||TS|| \in \{2, 8, 32, 128\}$ and was then tested on a different set of instances of size 1000. A record of the incorrect rankings produced was kept and used to calculate an error rate for each $< model, Q, ||TS|| >$ triplet. A more detailed description and a list of the evaluation functions tested is in the appendix.

The testing confirmed two expected behaviors. For all models and evaluation functions, the error rate dropped with increasing $||TS||$. For each evaluation function, eliminating errors of small magnitude from the error rate calculation did not change the relative accuracy ranking of the three models.

We found that for all evaluation functions, CIBL had a lower error rate than 1ARC. The largest difference is shown in Figure 4 for Q_3, a sinusoid, at $||TS|| = 128$:

$$ErrorRate_{CIBL} = 16.5$$

$$ErrorRate_{1ARC} = 22.5$$

On these data sets, CIBL's strategy of composing multiple exemplars is clearly superior to 1ARC's traditional instance-based approach.

We expected ID3 to perform better than CIBL on "smooth" evaluation functions, *e.g.,* those that have no change in the sign of their derivative. Indeed, ID3 outperformed CIBL for the planes Q_1 and Q_5, and the exponential Q_6. A surprise was that ID3 and CIBL performed equally on the folded plane Q_7, a function with an abrupt change in the sign of its derivative. We

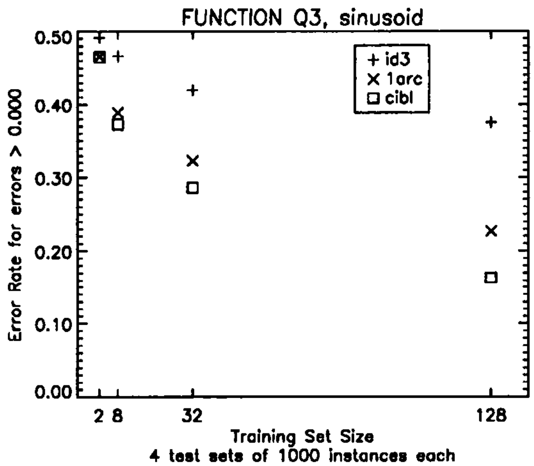

Figure 4: Accuracy of Q_3

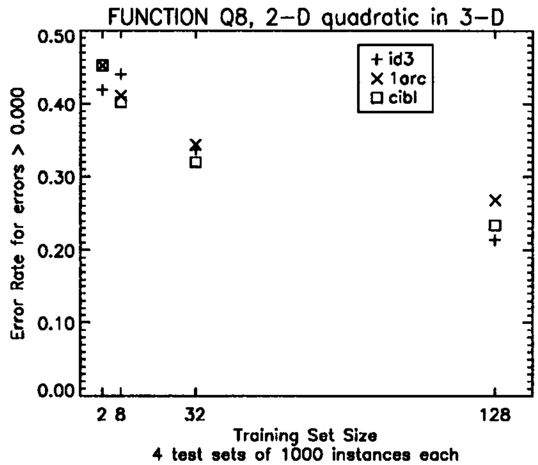

Figure 6: Accuracy of Q_8.

found that both 1ARC and CIBL were significantly better than ID3 on the quadratic Q_2 (Figure 5), the sinusoidal Q_3 (Figure 4), and the crossed planes Q_4. These functions exhibit changes in the sign of their derivatives in the form of local extrema or a severe discontinuity.

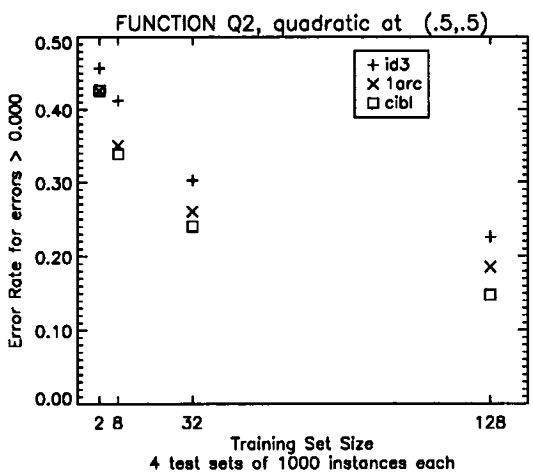

Figure 5: Accuracy of Q_2.

As expected, the addition of an irrelevant feature to the feature space—a feature that has no effect on the evaluation function—did not affect ID3's performance. CIBL's accuracy was degraded by the addition of an irrelevant feature, as shown in the testing data on Q_8 in Figure 6. Q_8 is the same 2-D quadratic as function Q_2, with an irrelevant third feature added. The problem is that CIBL's euclidean distance metric, used to assign costs to uncertain arcs, counts all features equally.

The sensitivity to irrelevant features exhibited by CIBL has been observed in other studies of instance-based learning [Aha, 1989]. Techniques for reducing this sensitivity were demonstrated by Aha and Goldstone [Aha and Goldstone, 1990]. However, composing exemplars tends to exacerbate the credit assignment problem of identifying low-relevance features.

The Impact of Representation on Preference Predicate Acquisition

The empirical evaluation indicates that the relative performance of instance-based and inductive approaches to preference predicate acquisition depends on the nature of the underlying quality function Q. If Q is irregular, *e.g.,* has discontinuities or local extrema, an instance-based approach such as CIBL is better, whereas if Q is smooth, an inductive approach is superior. However, the nature of Q depends critically on the representation of instances.

If instances are represented in terms of raw observables, any quality function on those instances is likely to be extremely irregular. For example, if chess positions are represented purely in terms of the locations of each piece on the board, the evaluation function will be extremely irregular, because changing the position of a single piece can drastically alter the evaluation of the position. If instances are represented in terms of *derived* or *abstract* features, however, the evaluation function may become much smoother. For example, if chess positions are represented in terms of strategic features such as control of files or pawn structure, or in terms of tactical features such as the existence of pins or the security of the king, an incremental change in a feature will usually change the evaluation function only incrementally. The ideal instance representation for the acquisition of preference predicates would include the quality function Q itself as a derived feature.[2]

Thus, a quality function that is a highly irregular when applied to a low abstraction representation may become smooth when applied to a higher abstraction representation. This suggests that the real issue in choosing between instance-based and inductive approaches to acquisition of preference predicates may be the nature of the representation of the instances. If instances are represented at a low level of abstraction, an instance-based approach like CIBL may be superior because of the irregularity of the quality function when applied to such descriptions. Induction may be more appropriate if instances are represented in terms of more abstraction features.

Related Work

Previous work addressing the composition of exemplars has focused on case fragments or "snippets." CADET [Sycara and Navinchandra, 1991] synthesized new designs by combining parts of multiple previous designs. Similarly, CELIA [Redmond, 1990] performed diagnosis using portions of multiple cases acquired through "explanatory apprenticeship." GREBE [Branting and Porter, 1991] generated legal arguments by combining portions of multiple legal precedents. CIBL differs from these approaches in that it composes entire exemplars rather than portions of exemplars. This reflects the fact that CIBL's exemplars—preference predicate instances—are far simpler than the cases in CADET, CELIA, or GREBE.

Instance-Based Learning vs. Case-Based Reasoning

The distinction between instance-based learning and case-based reasoning is relatively ill-defined. In general, however, instance-based learning can be viewed as a specialization of case-based reasoning typified by the following characteristics:

- Exhaustive matching rather than indexing.
- No case adaptation.
- Instance representation in terms of attribute-value pairs rather than complex structures.

Instance-based learning systems exhibiting these characteristics include MBRtalk [Stanfill and Waltz, 1986] and Aha's IB1 – IB4 [Aha and Goldstone, 1990].

Case-based reasoning research has typically emphasized knowledge-intensive approaches to indexing (*e.g.*, remindings compiled from explanations in Protos [Porter *et al.*, 1990] and the complex episodic memory organization developed in Cyrus [Kolodner, 1984]), adaptation (*e.g.*, operator sequence reuse in derivational analogy [Carbonell, 1986] and in CHEF [Hammond, 1986]) and complex case representations (*e.g.*, the relational representations used in GREBE [Branting, 1991] and most case-based design and planning systems).

1ARC is a typical instance-based learning algorithm. CIBL, by contrast, differs from typical instance-based learning algorithms in that it performs case adaptation. Rather than directly applying the solution (*i.e.*, ordering) in a previous instance to a new instance, CIBL composes multiple previous solutions to find a solution to a new case. However, the current implementation of CIBL uses an attribute/value pair representation and does not perform indexing.

Future Work

Our preliminary investigation of CIBL suggests a number areas for future investigation:

- *Evaluation using actual astronomical data.* The relative performance of CIBL and inductive approaches to learning preference predicates remains to be empirically assessed with actual astronomical scheduling data. The completion of the Observing Assistant should make such data available.
- *Similarity metric adaptation.* CIBL's euclidean distance metric rests on the assumption that all attributes are equally important. Matching would be improved if the distance function were adjusted based on ranking accuracy.
- *Noise handling.* The current evaluation assumed noise-free data. It may be possible to handle noisy data by assigning nonzero cost to training arcs that participate in incorrect rankings.
- *Alternative representations.* Instances in many domains may be desribed in terms of symbolic, rather than real-valued, features. To be useful in such domains CIBL needs to be extended to handle such features.
- *Comparison with other learning algorithms.* Additional approaches to acquiring preference predicates to which CIBL could be compared include a k-NN version of 1ARC (where the ordering of two states is determined from the k nearest arcs for $k > 1$) and backpropagation.

Conclusion

This paper has presented a strategy for learning preference predicates that permits multiple exemplars to be composed. This strategy permits the transitivity of P_Q to be directly exploited. An empirical evaluation with artificial data showed that CIBL was significantly more accurate for all evaluation functions tested than 1ARC, an IBL strategy unable to compose exemplars. In the absence of irrelevant features, both CIBL and 1ARC outperform ID3 when the underlying evaluation

[2] A well-known illustration of the dependence of inductive learning techniques on the representation of instances is Quinlan's experience that devising a set of derived features for chess board positions sufficient to enable ID3 to induce a decision tree for "lost in 3-ply" required 2 person-months [Quinlan, 1983].

function contains one or more extrema or a severe discontinuity. In the absence of extrema or discontinuities or when irrelevant features are present, ID3 outperforms the instance-based approaches.

Acknowledgements

This research was supported in part by a NASA Space Grant Graduate Fellowship given through the Wyoming Planetary and Space Science Center. We would like to thank Peter Turney for a number of useful suggestions concerning the application of ID3 to real-valued data.

Appendix: Details of Testing

Functions Tested

1x10 plane

$$Q_1(f_1, f_2) = f_1 + 10f_2$$

quadratic at (0.5,0.5)

$$Q_2(f_1, f_2) = (f_1 - 0.5)^2 + (f_2 - 0.5)^2$$

2-cycle sinusoid

$$Q_3(f_1, f_2) = sin(2\pi(f_1 + f_2))$$

crossed planes

$$Q_4(f_1, f_2) = \begin{cases} f_2 & \text{if } f_1 \leq 0.5 \\ 1 - f_2 & \text{otherwise} \end{cases}$$

1x1 plane

$$Q_5(f_1, f_2) = f_1 + f_2$$

exponential

$$Q_6(f_1, f_2) = exp(f_1^2 + f_2^2)$$

folded plane

$$Q_7 7(f_1, f_2) = \begin{cases} f_1 + f_2 & \text{if } f_1 \leq 0.5 \\ 1 + f_2 - f_1 & \text{otherwise} \end{cases}$$

2-D quadratic in 3-D

$$Q_8(f_1, f_2, f_3) = (f_1 - 0.5)^2 + (f_2 - 0.5)^2$$

Testing Notes

Each test triplet, $< model, Q, ||TS|| >$, was repeated with four different training sets and four different testing sets containing 1000 instances each. Error rates were computed by counting the errors from the four tests and dividing by 4000.

ID3 cannot learn training arcs directly. The relevant properties of a training arc—the location of the head, the location of the tail, the normalized direction, and the magnitude—must be encoded as a feature vector of real values as shown below. For each training instance $P_Q(H,T)$, ID3's training set contained a positive and a negative instance of the concept "is preferred to" as shown below.

$$< +, T, H, (H - T)/||H - T||, ||H - T|| >$$
$$< -, H, T, (T - H)/||H - T||, ||H - T|| >$$

References

Aha, D. W. and Goldstone, R. L. 1990. Learning attribute relevance in context in instance-based learning algorithms. In *Twelfth Annual Conference of the Cognitive Science Society.* 141–149.

Aha, D. 1989. Incremental, instance-based learning of independent and graded concepts. In *Proceedings of the Sixth International Workshop on Machine Learning.* 387–391.

Aha, D. 1990. *A Study of Instance-Based Algorithms for Supervised Learning Tasks.* Ph.D. Dissertation, University of Claifornia at Irvine.

Aha, D. 1992. Generalizing from case studies: A case study. In *Proceedings of the Ninth International Workshop on Machine Learning.* 1–10.

Aho, A.; Hopcroft, J.; and Ullman, J. 1974. *The design and Analysis of Computer Algorithms.* Addison-Wesley Publishing Co.

Branting, L. K. and Porter, B. W. 1991. Rules and precedents as complementary warrants. In *Proceedings of Ninth National Conference on Artificial Intelligence*, Anaheim. AAAI Press/MIT Press.

Branting, L. K. 1991. *Integrating Rules and Precedents for Classification and Explanation: Automating Legal Analysis.* Ph.D. Dissertation, University of Texas at Austin.

Callan, J.; Fawcett, T.; and Rissland, E. 1991. Adaptive case-based reasoning. In *Proceedings of the Third DARPA Case-Based Reasoning Workshop.* Morgan Kaufmann. 179–190.

Carbonell, J. G. 1986. Derivational analogy: s theory of reconstructive problem solving and expertise acquisition. In Carbonell, J. G.; Michalski, R. S.; and Mitchell, T. M., editors 1986, *Machine Learning*, volume 2. Tioga, Palo Alto, CA. 371–392.

Hammond, K. 1986. *Case-Based Planning: An Integrated Theory of Planning, Learning, and Memory.* Ph.D. Dissertation, Yale University.

Kolodner, J. 1984. *Retrieval and Organizational Strategies in Conceptual Memory: a Computer Model.* Lawrence Erlbaum Associates, Hillsdale, NJ.

Mitchell, T.; Mahadevan, S.; and Steinberg, L. 1985. Leap: A learning apprentice for vlsi design. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence.* Morgan Kaufmann.

Porter, B. W.; Bareiss, E. R.; and Holte, R. C. 1990. Concept learning and heuristic classification in weak-theory domains. *Artificial Intelligence Journal* 45(1–2).

Quinlan, J. R. 1983. Learning efficient classification procedures and their application to chess end games. In Carbonell, J. G.; Michalski, R. S.; and Mitchell, T. M., editors 1983, *Machine Learning*, volume 1. Tioga, Palo Alto, CA. 463–482.

Quinlan, J. R. 1986. Induction of decision trees. *Machine Learning* 1:81–106.

Redmond, M. 1990. Distributed cases for case-based reasoning; facilitating use of multiple cases. In *Proceedings of AAAI-90*, Boston. American Association for Artificial Intelligence.

Stanfill, C. and Waltz, D. 1986. Toward memory-based reasoning. *Communications of the ACM* 29(12).

Sycara, K. and Navinchandra, D. 1991. Influences: a thematic abstraction for creative use of multiple cases. In *Proceedings of the Third DARPA Case-Based Reasoning Workshop*. Morgan Kaufmann. 133–144.

Utgoff, P. and Clouse, 1991. Two kinds of training information for evaluation function learning. In *Proceedings of Ninth National Conference on Artificial Intelligence*, Anaheim. AAAI Press/MIT Press. 596–600.

Utgoff, P. and Saxena, S. 1987. Learning a preference predicate. In *Proceedings of the Fourth International Workshop on Machine Learning*. 115–121.