Using a Genetic Algorithm to Learn Prototypes for Case Retrieval and Classification

David B. Skalak
Department of Computer Science
University of Massachusetts
Amherst, MA 01003
skalak@cs.umass.edu

Abstract*

We describe how a genetic algorithm can identify prototypical examples from a case base that can be used reliably as reference instances for nearest neighbor classification. A case-based retrieval and classification system called **Off Broadway** implements this approach. Using the Fisher Iris data set as a case base, we describe an experiment showing that nearest neighbor classification accuracy of over 95% can be achieved with a set of prototypes that constitute less than 5% of the case base.

Introduction

One of the fundamental problems of case-based reasoning (CBR) is to ensure that exhaustive examination of every case in case memory need not be performed to retrieve the most similar or the most relevant cases. A variety of indexing and other strategies have been brought to bear successfully on this problem.

In particular, nearest neighbor case retrieval and classification systems have dealt with the problem of exhaustive case comparison. One class of solutions is to pre-process cases into data structures that enable fast nearest neighbor retrieval, such as Voronoi diagrams [Preparata & Shamos, 1985] or *kd*-trees [Moore, 1990]. An alternative approach is to perform exhaustive comparison, but to use parallel, distributed memory hardware [Stanfill & Waltz, 1986]. For tasks that do not require that all instances be stored in case memory, a third alternative is to reduce the number of reference instances in the case base[1].

Reducing the number of instances used for nearest neighbor retrieval has been a topic of research in the pattern recognition and instance-based learning communities for some time, where it is sometimes referred to as the "reference selection problem." Approaches to the problem have included storing misclassified instances (e.g., the Condensed Nearest Neighbor algorithm [Hart, 1968], the Reduced Nearest Neighbor algorithm [Gates, 1972], IB2 [Aha, 1990]); storing only training instances that have been *correctly* classified by other training instances [Wilson, 1972]; exploiting domain knowledge [Kurtzberg, 1987]; and combining these techniques [Voisin & Devijver, 1987]. (For discussions of these approaches, see generally [Aha, 1990].) For example, for numeric prediction tasks, Aha's IB2 (also called CBL2) algorithm saves only those training instances whose prediction error is above a given tolerance threshold [Aha, 1990]. Other systems deal with reference selection by storing averages or abstractions of instances. In this paper we describe a novel approach to reference selection: apply a genetic algorithm to identify small sets of instances that may be used as references for nearest neighbor retrieval.

This paper is an interim report on on-going research in which we describe a case retrieval and classification system called **Off Broadway.** We present an experiment on the Fisher Iris data set [Fisher, 1936], [Murphy & Aha, 1992] showing that Off Broadway achieves classification accuracy of over 95% with fewer than eight reference instances. Related research is discussed, and proposed work and a summary end the paper.

Off Broadway

Off Broadway is a case retrieval and classification system that performs classification based on a 1-nearest neighbor algorithm. The system attempts to learn a set of distinguished cases *(prototypes)* that have demonstrated classification power.

The system maintains a population of sets of potentially prototypical cases, where each prototype set is a subset of a fixed cardinality of the case base. Each set of prototypes is evaluated by its classification accuracy on a set of training cases, where the class of each training case is compared with the class of the prototype that is its nearest neighbor in a prototype set. A genetic algorithm is used to search the space of prototype sets in order to find a set with superior classification accuracy. We show that sets of prototypes can evolve whose classification performance on the Iris data set have success rates comparable to reported nearest neighbor algorithms that use much larger subsets of the data set.

* This research was supported in part by the National Science Foundation, contract IRI-890841, and the Air Force Office of Sponsored Research, contract 90-0359.

[1] Legal argument is an example of a task for which it would be dangerous to eliminate any case from memory.

The connotation of "prototype" suggests one of a small number of distinguished instances. For example, the number of leading cases on a particular legal issue is usually very small, often just one or two cases. Our immediate focus, therefore, is on identifying a very small number of cases as prototypes, and we arbitrarily look to designate fewer than 5% of the Iris data base as classification prototypes. Two general benefits of using nearest neighbor classification prototypes as surrogates for a much larger case set are obvious:

- decreased time to perform classification, since classification involves comparison with only a handful of prototypical cases; and
- decreased memory requirements, since only the prototypes need be stored in short-term memory.

One specific benefit of our approach is that no expert domain knowledge is needed to identify prototypical cases. Case-based reasoning often is used in domains lacking strong, operational domain theories, and so the absence of reliance on domain knowledge is an important aspect of our approach.

Genetic Algorithms

Genetic algorithms (GAs) are a class of adaptive search techniques that have often proven effective for searching large search spaces [Goldberg, 1989]. GAs maintain a population of members, usually called "genotypes" and classically represented by binary string, which can be mutated and combined according to a measure of their worth or "fitness", as measured by a task-dependent evaluation function. As described in [Holland, 1986], the basic execution cycle of a typical GA is straightforward:

1. Select pairs of population members from the population according to fitness, so that stronger members are more likely to be selected.
2. Apply genetic operators to the pairs, creating offspring. Random mutation of a population member is a typical operator. Another genetic operator, "crossover" exchanges a random segment between two members of the population.
3. The members of the population with the lowest fitness are replaced by the offspring.
4. Return to 1. unless certain termination criteria have been satisfied, such as the creation of an individual whose fitness meets some threshold, or the stabilization of the population.

In this application, the search space is the set of all subsets of a fixed cardinality of a case base. If n prototypes are selected from a case base of m cases, there are $C(m, n)$, "m choose n", possible sets of prototypes. For our experiments using the Iris data set, m=120 (30 cases are reserved for testing) and $n \leq 8$, so the search space is still quite large.

While GAs have been used in the past for rule-based classification (e.g., [Holland, 1986], [De Jong, 1990], [De Jong & Spears, 1991]), there has been recent interest in exemplar-based classification techniques using GAs [Kelly & Davis, 1991] as well. De Jong and Spears point out that the two concept description languages in general use in machine learning are decision trees and rules. The research in this paper investigates a third description language that is symptomatic of exemplar-based approaches to classification: the cases themselves. Our approach is an example of the "single representation trick" applied elsewhere in machine learning (see [Barr, et al., 1981]), in which instances and instance generalizations are expressed in the same language (e.g., [Michalski & Chilausky, 1980]). However, we use this trick in reverse: here the concepts are expressed in the language of cases, rather than expressing instances in the generalization language.

We see several advantages to this exemplar-based approach to concept description. Given a fixed case base, the set of concept descriptions is finite, and therefore possibly more tractable than description languages that admit infinitely many concept descriptions. Second, there is minimal bias in the concept representation language of the cases: the only bias is implicit in the cases that have already been exposed to the system through inclusion in the case base. We view the presence of minimal bias as an advantage supporting wider applicability of this approach, although we speculate with [De Jong & Spears, 1991] that performance may suffer on tasks that are amenable to some *a priori* bias in the concept description language.

Details of the Genetic Algorithm to Identify Sets of Prototype Cases

Basic Approach: In general terms, Off Broadway's GA is a generational genetic algorithm that uses uniform crossover and a stochastic remainder without replacement selection procedure. The GA maintains a population of individuals, where each individual is a set of n prototypes. Each prototype is a member of a fixed case base. Here we report on experiments in which n ranges from 3 to 8, which represent prototype sets constituting less than approximately 5% of the entire case base. The fitness of each member of the population is its classification accuracy. Our basic approach is analogous to the Pittsburgh ("Pitt") approach to optimizing rule sets [Smith, 1983] in that organisms in the population are *sets* of prototypes, rather than individual prototypical cases.

Encoding: Each prototype set is encoded as a binary string, which is conceptually divided into n substrings, one for each of the n prototypes, where each substring encodes an index into a case base stored as an array (Figure 1). The length of the binary string encoding a prototype set is the number of bits required to represent the largest index of a case in the case base, multiplied by the number of prototypes, $\lceil \log_2 m \rceil * n$, where m is the number of cases in the case base.

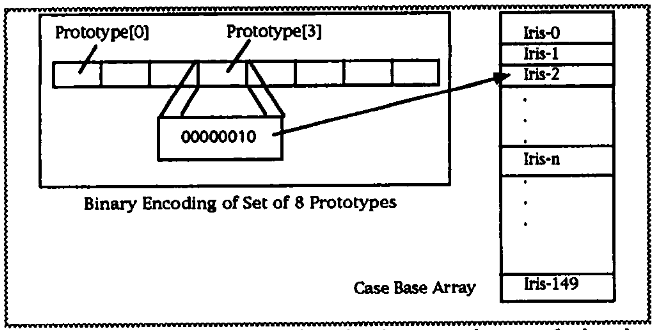

Figure 1. Each organism in the genetic population is a binary encoding of a set of indexes into a case base array.

Evaluation Function: The fitness of each set of prototypes in the population is determined by its classification accuracy, specifically, the percentage of the training set of cases that it correctly classifies. A training case is correctly classified by a prototype set if the training case's class equals the class of the prototype that is its 1-nearest neighbor in the prototype set.

The similarity function used in this nearest neighbor computation is simple, particularly since the Iris cases are represented by numerical features. The raw values for each case feature are linearly scaled from 0 to 100, where any value less (greater) than three standard deviations from the mean of that feature's value across all cases is assigned the value 0 (100). To compute the similarity distance between two cases, we first calculate the feature-by-feature difference of two cases with these scaled values. The unweighted arithmetic average of such differences over all the features of a case is the similarity distance between the two cases. The prototype with the smallest such similarity distance to a test case is its nearest neighbor. The ReMind case-based reasoning development shell has previously incorporated a similar approach to nearest neighbor similarity assessment [Cognitive Systems, 1992].

Initialization: The population is initialized to a random population of 20 prototype sets.

Operators: Mutation and crossover operators effectively alter the set of prototypes by changing the index of one or more members of each prototype set. Crossover splits may occur anywhere within an organism. The probability of mutating an organism (not a single locus) was fixed at 0.03. The crossover probability was set at 0.7 .

Off Broadway is implemented in Macintosh Common Lisp v.2.0 using CLOS. Part of the system instantiates a shell for creating GA applications available from GNU software [Williams, 1992].

We now turn to an experiment to identify prototypes for classification.

Experiment: Classification by Off Broadway

We have performed a set of experiments to demonstrate the identification of prototypes by GA using the UCI Iris data set of 150 Iris types [Murphy & Aha, 1992]. Each Iris case has four rational-number features (sepal length, sepal width, petal length, petal width) and is assigned one of three classifications (Iris-setosa, Iris-versicolor, Iris virginica.) The database contains 50 cases of each classification. An example of an Iris case is

```
(5.1 3.5 1.4 0.2 Iris-setosa).
```

Our experimental methodology was to perform 5-fold cross validation of the classification performance of Off Broadway. The case base was randomly partitioned into 5 groups of 30 cases. For each validation, we used one of the partitions of 30 random cases as testing instances and reserved the remaining 120 instances for training. Prototypes were selected only from the training sets and were used to classify the 30 test instances. We tested the classification accuracy of sets of *n* prototypes determined by Off Broadway, where *n* varied from 3 to 8. Our lower limit was 3 prototypes in these experiments, since the Iris data set contains three classes of Iris types.

For each partition, the GA was run until it performed 100 evaluations of the training set, approximately 6 generations on average, taking approximately 10 minutes on average on an Apple Macintosh IIx.

Iris cases were stored in a case array in the order in which they appear in the UCI Iris database file, where they are grouped by class. Recall from the description of the algorithm that the order of the stored cases may be relevant because the members of the GA population encode indexes into this case array.

We were primarily interested in the classification accuracy of the best performing member of a population. Average best performance for the learned prototype systems was computed using the following procedure.

a. With the number of prototypes in each prototype set fixed, we divided the cases base into five random partitions, each consisting of a 120-element training set and a 30-element test set, as described in the second paragraph in this section.

b. For each of the five partitions of the case base, we determined the *maximum* percentage of the 30 test cases correctly classified by an individual in the population of 20 prototype sets after 100 evaluations of the population.

c. Finally we computed the average of the resulting 5 *maxima*. This average is reported in the second column of Table 1, labeled "GA Prototypes Ave. Max. Correct".

Prototypes (n)	GA Prototypes Ave. Max. Correct (%)	
3	28.6	(95.3%)
4	29.2	(97.3%)
5	28.6	(95.3%)
6	29.0	(96.7%)
7	29.2	(97.3%)
8	28.8	(96.0%)

Table 1. Average maximum performance for learned prototype sets, using test sets of 30 instances.

The basic result, which is reported in Table 1, shows that the small number of prototypes identified by Off Broadway performed with greater than 95% classification accuracy in the 5-fold cross validation experiments using the Iris data set. The data in Table 1, which show the average *best* performance of a member of the final population, reflect how the algorithm would probably be used in practice: a best performing member of the population would be identified and used as a surrogate for the entire case base.

These results are comparable to or better than a number of reported nearest-neighbor methods. [Weiss & Kapouleas, 1989], for example, report a correctness of 96% for nearest neighbor retrieval on the Iris data set, using a leaving-one-out evaluation methodology. Our approach performed significantly better than the 76.6% correct given in [Fertig & Gelernter, 1991], which in turn performed slightly better than the "neighborhood census rule" described in [Dasarathy, 1980]. In one run, [Gates, 1972] reported 100% classification accuracies using the condensed nearest neighbor rule and the reduced nearest neighbor rule, which used 20 and 15 reference instances, respectively. It is unclear whether cross-validation or some other experimental test was performed to confirm this result, however. By pre-processing the raw Iris data and using a 1-nearest neighbor algorithm, Gates also reported classification accuracies from 93.3% to 96.7% using from 15 to 31 reference instances. Finally, using from 3 to 6 "pathological" reference instances, Gates found classification accuracy varied from approximately 17% to 40%. See [Gates, 1972] for descriptions of these and related experiments.

Limitations of Off Broadway

The current implementation is limited by the assumption that the number of prototypes desired is fixed *a priori*, in advance by the user, and is supplied as a parameter to the system. However, since GAs using the Pitt approach have explored variable-length rule sets, we believe that a fairly minor re-implementation of the system would dispense with this assumption.

One important limitation of the research reported here lies with the simplicity of the Iris data set, and we plan to test the approach on more complex data sets. In the Iris data set, one class is linearly separable from the other two, but the other two are not separable from each other. We have begun to gather evidence that shows, in fact, that sets of prototypical instances may be relatively easy to find in the Iris domain.

Other data sets may also include elements that are more clearly prototypical, as that term is usually used. For example, the LED-7 display domain for light-emitting diodes [Murphy & Aha, 1992], contains obvious prototypical elements, the correct numerals themselves[2].

Related Research

GA classification systems have been created by [Kelly & Davis, 1991] to learn real-valued weights for features in a data set and by [De Jong & Spears, 1991] to learn conceptual classification rules. Our approach is similar to De Jong and Spears's in that they used a Pitt approach to defining their population, and applied the same straightforward fitness function for each individual in their population, the percentage correct on a set of training examples. However, an important distinction with this work is that De Jong and Spears used classification rule sets as population individuals, rather than prototypical cases.

Protos [Bareiss, 1989] is a good example of a case-based reasoning system that relies on case prototypes for classification. An exemplar that is successfully matched to a problem has its prototypicality rating increased. The prototypicality rating is used to determine the order in which exemplars are selected for further, knowledge-based pattern matching. Protos is an intelligent classification assistant and the prototypicality rating may be increased based in part on the actions of the supervising teacher. The ReMind case-based reasoning development shell [Cognitive Systems, 1992] also incorporates a facility for the user to select prototypes to further index a case base.

Approaches to reducing the number of reference instances for nearest-neighbor retrieval were discussed generally in the introduction.

This line of research is an immediate outgrowth of two projects from our group. In [Skalak & Rissland, 1990] we suggested and evaluated a case-based approach to the problem of reducing the number of training instances needed to create a decision tree using ID5 [Utgoff, 1988]. Recently we described a system called **Broadway** [Skalak, 1992] that represents cases as blackboard knowledge sources whose preconditions invoke local similarity functions that apply only within a closed neighborhood of each case in the space of cases. The connection between the current project, Off Broadway, and Broadway is that the set of prototypes learned by Off Broadway tessellates the case space into local neighborhoods (as for any nearest neighbor algorithm), where each neighborhood contains the volume of cases in case space for which a particular prototype is the nearest neighbor.

Future Work and Summary

A next step in the Off Broadway system is to learn what similarity metric applies in each region in which a specific prototype is the nearest neighbor. We are investigating an algorithm that uses a version of the absolute error correction rule [Nilsson, 1990] together with a state preference method [Utgoff & Clouse, 1991] to learn a similarity function for each such region.

In summary, Off Broadway attempts to learn sets of case classification prototypes using a genetic algorithm.

[2] "The problem of reading *printed* characters is a clear-cut instance of a situation in which the classification is based ultimately on a fixed set of 'prototypes'." [Minsky, 1965, p. 21].

We have applied Off Broadway to classify elements of the Iris database and have shown that classification performance better than or equal to that achieved using much larger reference sets can be achieved using fewer than 8 Iris cases as classification prototypes.

Acknowledgments

This work has benefited from discussions with Edwina L. Rissland, Jeff Clouse, Claire Cardie and Timur Friedman.

References

Aha, D. W. (1990). *A Study of Instance-Based Algorithms for Supervised Learning Tasks: Mathematical, Empirical, and Psychological Evaluations*. Ph.D. Thesis, Available as Technical Report 90-42, Dept. of Information and Computer Science, University of California, Irvine, CA.

Bareiss, E. R. (1989). *Exemplar-Based Knowledge Acquisition*. Boston, MA: Academic Press.

Barr, A., Feigenbaum, E. A. & Cohen, P. (1981). *The Handbook of Artificial Intelligence*. Reading, MA: Addison-Wesley.

Cognitive Systems, Inc. (1992). *ReMind: Case-based Reasoning Development Shell*. New Haven, CT.

Dasarathy, B. V. (1980). Nosing Around the Neighborhood: A New System Structure and Classification Rule for Recognition in Partially Exposed Environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(1), 67-71.

De Jong, K. (1990). Genetic-Algorithm-Based Learning. In Y. Kodratoff, & R. Michalski (Eds.), *Machine Learning* (pp. 611-638). San Mateo, CA: Morgan Kaufmann.

De Jong, K. A. & Spears, W. M. (1991). Learning Concept Classification Rules Using Genetic Algorithms. *Proceedings, 12th International Joint Conference on Artificial Intelligence*, 651-656. Sydney, Australia. International Joint Conferences on Artificial Intelligence.

Fertig, S. & Gelernter, D. H. (1991). FGP: A Virtual Machine for Acquiring Knowledge from Cases. *Proceedings, 12th International Joint Conference on Artificial Intelligence*, 796-802. Sydney, Australia. International Joint Conferences on Artificial Intelligence.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annual Eugenics*, 7, 179-188.

Gates, G. W. (1972). The Reduced Nearest Neighbor Rule. *IEEE Transactions on Information Theory*, (May), 431-433.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA: Addison-Wesley.

Hart, P. E. (1968). The Condensed Nearest Neighbor Rule. *IEEE Transactions on Information Theory (Corresp.)*, IT-14(May), 515-516.

Holland, J. H. (1986). Escaping brittleness: The possibilities of general purpose learning algorithms applied to parallel rule-based systems. In *Machine Learning*. San Mateo, CA: Morgan Kaufmann.

Kelly, J. D. J. & Davis, L. (1991). A Hybrid Genetic Algorithm for Classification. *Proceedings, 12th International Joint Conference on Artificial Intelligence*, 645-650. Sydney, Australia. International Joint Conferences on Artificial Intelligence.

Kurtzberg, J. M. (1987). Feature analysis for symbol recognition by elastic matching. *International Business Machines Journal of Research and Development*, 31, 91-95.

Michalski, R. S. & Chilausky, R. L. (1980). Learning by being told and learning from examples: An experimental comparison of the two methods of knowledge acquisition in the context of developing an expert system for soybean disease diagnosis. *International Journal of Policy Analysis and Information Systems*, 4, 125-161.

Minsky, M. (1965). Steps Toward Artificial Intelligence. In R. D. Luce, R. R. Bush, & E. Galanter (Eds.), *Readings in Mathematical Psychology*, vol. II., 18-40. New York: John Wiley.

Moore, A. W. (1990). Acquisition of Dynamic Control Knowledge for a Robot Manipulator. *Proceedings, Seventh International Conference on Machine Learning*, 244-252. Austin, TX. Morgan Kaufmann.

Murphy, P. M. & Aha, D. W. (1992). *UCI repository of machine learning databases*. Irvine, CA: University of California, Dept. of Information and Computer Science.

Nilsson, N. J. (1990). *The Mathematical Foundations of Learning Machines*. San Mateo, CA: Morgan Kaufmann.

Preparata, F. P. & Shamos, M. I. (1985). *Computational Geometry: An Introduction*. New York: Springer-Verlag.

Skalak, D. B. (1992). Representing Cases as Knowledge Sources that Apply Local Similarity Metrics. *Proceedings, The 14th Annual Conference of the Cognitive Science Society*, 325-330. Bloomington, Indiana. Lawrence Erlbaum.

Skalak, D. B. & Rissland, E. L. (1990). Inductive Learning in a Mixed Paradigm Setting. *Proceedings of AAAI-90, Eighth National Conference on Artificial Intelligence*, 840-847. Boston, MA. American Association for Artificial Intelligence.

Smith, S. F. (1983). Flexible Learning of Problem Solving Heuristics Through Adaptive Search. *Proceedings, 8th International Joint Conference on Artificial Intelligence*, 422-425. Karlsruhe, West Germany. International Joint Conferences on Artificial Intelligence.

Stanfill, C. & Waltz, D. (1986). Toward Memory-Based Reasoning. *Communications of the ACM*, 29(12), 1213-1228.

Utgoff, P. E. (1988). ID5: An Incremental ID3. *Proceedings of the Fifth International Conference on Machine Learning*. Ann Arbor, MI.

Utgoff, P. E. & Clouse, J. A. (1991). Two Kinds of Training Information for Evaluation Function Learning. *Proceedings, AAAI-91*, 596-600, AAAI Press/MIT Press.

Voisin, J. & Devijver, P. A. (1987). An application of the Multiedit-Condensing technique to the reference selection problem in a print recognition system. *Pattern Recognition*, 20(5), 465-474.

Weiss, S. M. & Kapouleas, I. (1989). An Empirical Comparison of Pattern Recognition, Neural Nets, and Machine Learning Classification Methods. *Proceedings, 11th International Joint Conference on Artificial Intelligence*, 781-787. Detroit, MI. International Joint Conferences on Artificial Intelligence.

Williams, G. P. W. Jr. (1992). *GECO: Genetic Evolution through Combination of Objects*. Free Software Foundation.

Wilson, D. (1972). Asymptotic Properties of Nearest Neighbor Rules using Edited Data. *Institute of Electrical and Electronic Engineers Transactions on Systems, Man and Cybernetics*, 2, 408-421.