From: AAAI Technical Report FS-95-04. Compilation copyright © 1995, AAAI (www.aaai.org). All rights reserved.

A Testbed For Experiments In Adaptive Memory Retrieval and Indexing

Liwu Chang & Patrick Harrison
Naval Research Laboratory
Washington, DC USA
liwu@aic.nrl.navy.mil

Abstract

In this paper, we discuss properties of the Knack system which is designed as a testbed for experimenting memory retrieval and organization in Case-Based Reasoning. Methods used for retrieval and indexing are based on mathematically sound techniques developed in classification, clustering and decision analysis. Retrieval is done using decision theoretic methods such as voting k nearest neighbor and Bayesian theory with weighted attributes. New indices for cases are generated by using clustering methods. Cases are then re-organized using the new indices. The Knack environment was designed so that additional techniques and metrics can easily be added.

Introduction

Case-based reasoning is a problem solving paradigm that uses previous experiences in problem solving (e.g., [Kolodner, 1993]). At its core, case-based reasoning has adaptive methods for memory indexing, retrieval and organization. Knack was designed as a testbed for studying these adaptive memory functions using real-world problems. The evolution of Knack was based on abstract, functional model of memory that provided a basis for Knack analogs of human capacities to organize, index and retrieve information [Harrison, Chang & Meyrowitz, 1994].

Adaptation is commonly viewed as a process to achieve optimum results. Contents of adaptive methods range from the control of reflexive responses to the refinement of knowledge bases. In this paper, we describe two aspects of adaptation which are relevant to retrieval and indexing in memory based reasoning. The first aspect of adaptation is about the dynamic updating of similarity measurements used in retrieval procedures. In Knack, the computation of similarity is like evidence combination that pools the measure of each attribute, where the measure can be the function of the difference between values of two instances at that attribute. Factors, such as the importance weights and dependency relationships of attributes, that have great impact on the computation are accounted for as the components of adaptation. In dynamic environments, the importance weights of attributes vary as new events occur. The computation of weight change is dealt by using the probabilistic network model. Consequently, the similarity measurements are also updated. The second aspect of adaptation is about the generation of indices for case organization. One purpose of organizing cases is to facilitate coherent retrieval and control search attention. Among many possible approaches, in Knackorganization is to group stored cases hierarchically. Generation of new indices is carried out using clustering methods. As a tool, we propose the integrated use of incremental and hierarchical clustering procedures. The context or perspective information may be incorporated in case organization. In our experiments, we account for the perspective information as latent factors, each corresponding to a subset of attributes. As a result, different cases can be retrieved from different perspectives.

Figure 1 shows the top level Knack interface. It provides a user interface designed for experimentation with the analogs mentioned above. It provides support for running multiple experiments and comparing results. Knack provides the means to implement models of memory and to do case-based reasoning. It is a testbed for testing concepts in retrieval and (re)-indexing. In Knack, the case indices are represented in the form of an attribute list. With this representation, we can use UCI data collection to test methods implemented for retrieval and (re)-indexing. We test the predictive accuracy of retrieval methods and the quality of indices generated in (re)-indexing.

In the next section, we describe similarity functions and decision functions used for retrieval. We also describe methods for weighting attributes, analyzing their dependency and evaluating the importance index of a single stored case. Methods for re-indexing and case organization are discussed in section four. A summary is given in the last section.

Decision-based Retrieval

Retrieval is an interactive decision-making process with the goal to select cases and rank them. Three

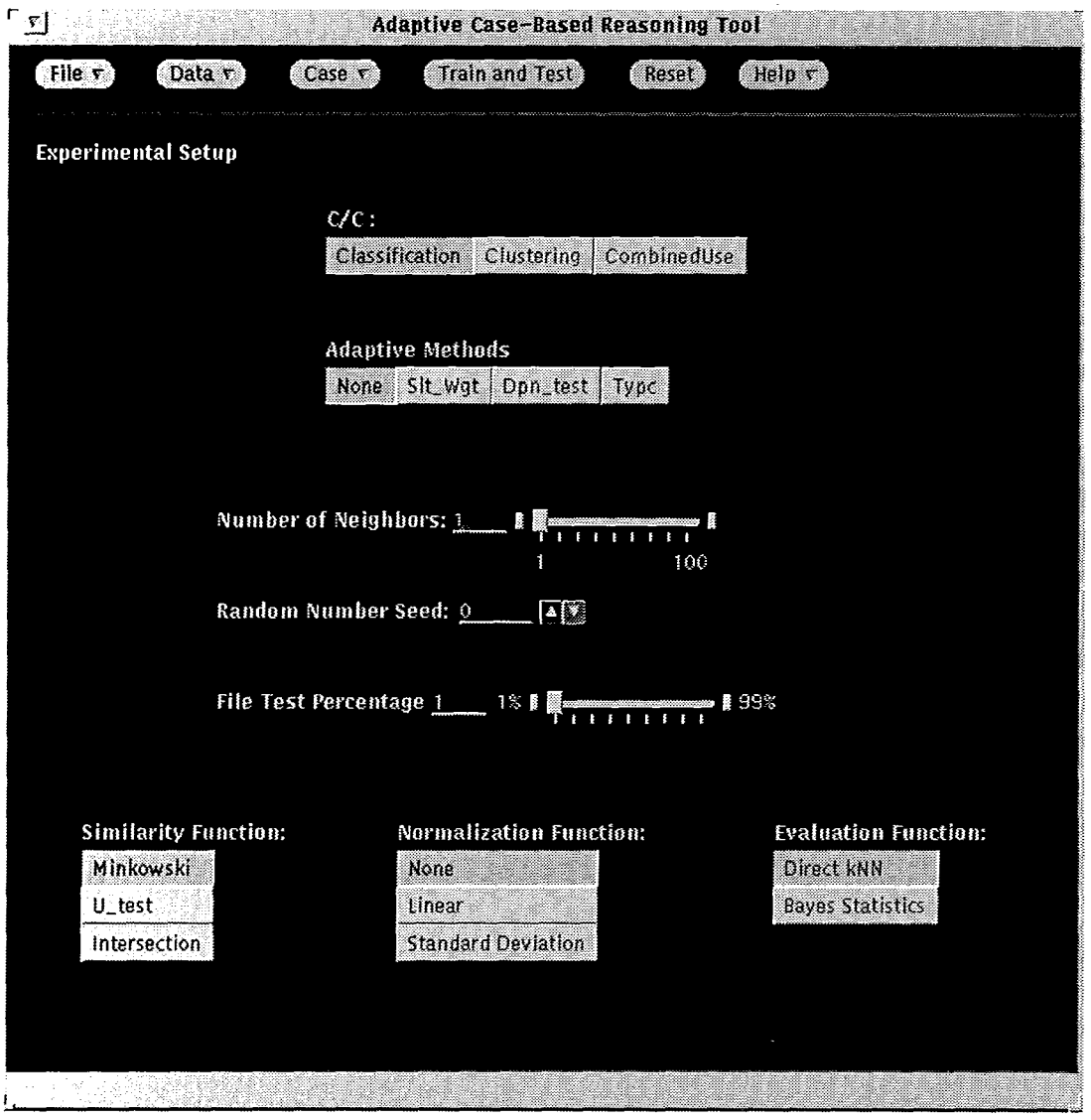

Figure 1: The Top Level Knack Interface

forms of retrieval are considered: First, given a target case, we might want a most similar case to be retrieved. In this situation, the index set of the target is compared against that of each stored case. Second, given an query, we might want a case that explains the query best. The query is encoded in terms of the indices and the decision is based on the encoded relations. Finally, we might want exemplar cases retrieved from a category according to information about that category. Decision functions such as Bayesian theory and k-NN are employed to handle different forms of retrieval. In this section, we will describe methods for decision based retrieval. Methods for selecting exemplar cases will be given in the next section.

Decision functions

Similarity in kNN is measured in terms of the Minkowski metrics for data of ordinal scale and higher. For nominal data, the measure of similarity is based on counting the number of matches. We also applied the similarity metric to computing the difference between two probability distributions.

For retrieval based on the measure of similarity between indices of target and stored cases, nonparametric decision procedure is useful. In Knack, the voting kNN, among other variations of kNN [Fukunaga, 1990][Wettschereck, 1995], is used as its statistical properties have been well studied. Roughly, voting kNN bases its decision on the frequency count of the majority class label seen in the first k nearest

neighbors. Its asymptotic error is calculated under the assumption that the probability of a particular class given the test data is the same as that of its nearest neighbor.

We may use Bayesian theory if the conditional probability distribution (referred to as c.p.d.) between queries and attributes is available. In Knack, c.p.d. decisions were carried out by using the probabilistic network model [Pearl, 1988]. For a simple two-layered network, a parent node stands for a query and each child node represents an attribute. In some situations, c.p.d. need be learned inductively from stored data. The c.p.d. is computed for each attribute separately. The estimation of c.p.d. is carried out via marginalization. Suppose the likelihood of stored data given a set of parameters obeys the multinominal distribution. By averaging over all possible parameters' values with the assumption of non-informative prior [Berger, 1985], the expected value of a parameter is estimated by the ratio

$$\frac{1 + n_{fvc}}{n_f + n_c} \tag{1}$$

where n_{fvc} is the number of data of cluster c at attribute f with value v, n_c stands for number of data in the cluster c and n_f stands for the total possible values of attribute f [Cooper & Herskovits, 1992]. This result can also be obtained from the EM algorithm [Dempster etc., 1977]. If n_c grows faster than n_f as more data arrive, the expected value will be close to the frequency count. For continuous data, c.p.d. is estimated using the Parzen approach [Parzen, 1962] with a normal kernel function.

Attribute values may be missing for some stored cases. The result of decision with missing data may be obtained from averaging over all possible values that the missing data can take. This approach becomes impractical when the possible values of attributes are large. In Knack, the unknown value of an attribute is viewed as a new category for the parametric approach and is ignored for the non-parametric approach. It may be that the attribute value of the input instance does not equal any existing values of that attribute when values of stored cases do not cover or we do not know its entire range of values. For data of ordinal scale, the unseen attribute value is added to the original set of values. An equal likelihood ratio is assumed to account for the ignorance for data of nominal scale.

Methods for calculating importance index

In a broad sense, weights of attributes, classes, temporal and spatial measurements, and instances all have impacts on the measure of similarity and consequently, the outcome of a decision. Among those weights, in Knack, we consider three types of importance indices that affect the decision analysis. The first type is related to selectively weighting attributes in the environment so that only important aspects will be attended. One weighting method is to evaluate the informativeness of attributes. The second type is the dependency relationship among attributes. A Bayesian model with independence assumption may yield inaccurate estimates if some attributes are indeed statistically dependent. The third type is related to the calculation of the importance index of each stored case. Cases with high ranking are retained or used for other applications (e.g., model design).

Relationships among attributes and cases can be inductively learned from stored cases. Learning methods differ according to their evaluation schemes and search strategies [Langley, 1994]. Two commonly applied evaluation schemes are filter and cross-validation. In the current version of Knack, a greedy search strategy is used for most situations. However, in dependency analysis due to the non-monotonic nature of probability computation, exhaustive search may be needed.

weighting attributes The key feature of the proposed approach to attribute weighting is by to evaluate the informativeness of each attribute. Weights are adjusted via training.

The informativeness is measured in two ways. First, weights are computed with c.p.d. and no cross-validation involved. An attribute has high discriminability if the c.p.d. (with respect to this attribute) is very dissimilar between classes. The proposed (dis)similarity metric is based on the measure of difference between two c.p.d. functions, i.e.,

$$\frac{1}{2}\sum_{j}^{J} |p(F_i = j \mid C_a) - p(F_i = j \mid C_b)|^2 \tag{2}$$

where F_i stands for the i^{th} attribute, j stands for the possible values of the i^{th} attribute and C is the class label. The overall discriminability of an attribute j is equal to the summation of the (dis)similarity measures for all pairs of classes. The equation reaches the minimum when two c.p.d. functions are identical and have the maximum value 1. Finally, the weight associated with an attribute can be obtained by homomorphically transforming the discriminability of feature j from [0,1] interval to the real number line.

The second approach is to select the subset of attributes via a cross-validation procedure. The selection process is carried out by comparing error rate of tests with different subset of attributes. Start with null set. One attribute is added to the attribute set at the end of each test run. Instead of randomly picking up an attribute, in Knack, the order of attributes to be selected is ranked according to the measure of informativeness. For each run, the training data is randomly divided into two parts. This division is repeated a number of times. The average correct test ratio of divisions is recorded and stands for the utility score of the subset of attributes under consideration. The final selection of the desired subset of attributes is the one with the highest scores. A percentage of the attributes will be

suppressed.

Suppression of non-informative attributes would generally improve predictive accuracy. For instance, our experiments with a letter-recognition test indicated that estimation, based on both kNN and Bayes analysis, using attributes with adjusted weights yields better results than those without weights. The improvement of error rate is around 9 percent on the average of 20 test runs with 1000 samples at 50 test percentage. The ranking of informativeness of attributes given by the proposed method and mutual information measure [Musman etc., 1993] is similar for this test. The proposed metric differs from mutual information, disorder measure of C4.5 [Quinlan, 1992] and value-difference metric [Stanfill & Waltz's, 1986] in that it is used not only for ranking but for weighting. Binary selection yields slightly better results than weighting method for this (around 0.8 percent) and some other tests but requires extremely longer training time. The reason for better outcomes may be that cross-validation takes into account several factors. A study of binary selection is given in [Aha & Bankert, 1994]. Incremental weight adjustment is not included in the current version of Knack.

analyzing dependency relationship The task is to discover a partition of attributes that best represents the distribution of stored data by exploring underlying dependencies among attributes. We consider two situations. In the first situation, each stored case is assumed to have a category label attached. Dependency among attributes is evaluated in terms of c.p.d given the class labels. For nominal-scaled data the proposed approach derives the dependent relationship of attributes on the basis of improved error rate of test results. The improvement is measured before and after attributes are joined. This approach iteratively joins together different attributes and selects the combination with the highest predicted accuracy. Search halts when the join does not yield better results. Unlike attribute selection, at each test run, a partition, rather than a subset, of attributes is under evaluation. Also unlike attribute selection, no priority is assigned to attributes. Hence, dependency analysis deals with larger search space. For ordinal and interval data, the Parzen approach is initiallu used to estimate probability density for each attribute. Following the same steps, we repeatedly evaluate with different partitions of the attributes.

In the second situation, suppose no information about category labels is available. We may employ a statistical test to test dependency when data is of nominal form. One example is Chi-Square with a given significance level. The joint probability of two sets of variables is assumed to be the expected value and the multiplication of each constituent probability density is assumed to be the observed value. The hypothesis is to test whether the two values fit well. The search strategy is the same as before. This approach is also applicable to the case when the category label is known. For ordinal data, the independence condition can be tested with selected areas or a number of sampled points rather than integration over entire space - integration is computationally expensive for high dimension data.

The approach for testing dependency via cross-validation is also discussed in [Pazzani, 1995]. Test results with some UCI data file showed improved predictive accuracy. For instance, test with congressional voting records yielded around 4 percent improvement on average with the parametric approach. However, the approach with joined attributes quickly takes up huge chunk of memory. Actually memory space required for a fully joined probabilistic distribution is exponential in the number of values of attributes. Effective ways of evaluating groups of attributes are currently being investigated.

calculating typicality This method differs because it assigns weights to each stored case or sample. The idea is based on the assumption that the data points of high typicality give better prediction. Therefore, a relatively small set of representatives are selected from a distribution and the reduced set hopefully maintains a similar predictability as the original set of cases. Typicality is also used as a criterion for selecting exemplars or representatives.

For the non-parametric approach, typicality of an instance is determined by the ratio of the number of times the instance is referenced (e.g., evaluated as the nearest neighbor to other points) and the number of times it is correctly referenced. With the parametric approach, typicality of an instance can be computed in terms of its membership. Suppose the instance is in class C_i, membership is the summation of the ratio between the likelihood of this instance and the maximum possible probability of C_i, i.e,

$$\frac{p(F_i = j \mid C_i)}{max(p(F_j \mid C_i))} \tag{3}$$

The test results (e.g., letter-recognition test) indicated that the error percentage almost remained unchanged (i.e., with small fluctuation) when data points with typicality measure less than 0.3 were removed using voting kNN. Although the measure of typicality discussed is computed via a filter scheme, it can be obtained with a cross-validation procedure. Typicality calculation in unsupervised learning selects the set of representatives with which properties of the original distribution could be maintained.

With the dependency analysis and weighting methods, we may construct belief network representation [Verma & Pearl, 1990] a decision tree model for better conceptual illustration of the relationships. For example, a new memory may be constituted in terms of a probabilistic network model describing relations between a query and a set of selected attributes. Since the importance index varies as new cases are present,

the similarity measurement will be updated and hence, the outcomes of retrieval will be different. The ability to update similarity measurement in changing environments is also one of the characteristics of the notion of dynamic memory [Schank, 1982].

Generation of indices

The task of organizing stored cases is to explore relationships among the set of attributes. The result may be a hierarchical representation of stored cases. The new structures can be described by using existing attributes or new terms. Vocabularies of new terms may be predefined. Case structure, like a discriminant network, may be indexed with existing weighted attributes. With case structures generated from unsupervised learning, new indices are needed for each cluster.

Among several approaches (e.g., [Ram & Santamaria, 1993]), in Knack, clustering methods are employed for the derivation of new indices. Case retrieval with new indices is expected to be more effective and coherent. Primarily, we consider clustering in a hierarchical form and a incremental form - the former deals with all data while the later evaluates one instance a time. The two forms can be used separately or combinatively. We applied the nearest neighbor principle to both forms as the criterion for combining two clusters. We consider not only the clustering form but also the quality of the methods (i.e., the ability of handling various distribution). In this section, we describe the proposed clustering method i.e., the combined use of hierarchical and incremental forms. Next, we describe a way of incorporating perspective information in case organization. The following discussion will focus on the parametric (c.p.d.) approach.

Proposed clustering methods for handling re-indexing

In the incremental form, the system determines whether the new sample should be assigned to one of the existing clusters or should be given a new label based on a cluster criterion g. The c.p.d. is initially null and is updated as new data comes in. Label of the new arrival is assigned based on posterior probabilities of candidate clusters. In Knack, the rejection criterion g is related to the maximum value of probability that a cluster C_j could have, i.e.,

$$\prod_i max(p(F_i \mid C_j)) \qquad (4)$$

Suppose the new arrival is assigned to cluster C_j. Criterion g measures the ratio between the maximum likelihood in the presence of the new data and the maximum possible probability that the cluster C_j could have. The idea is that if the arriving data is truly desirable, the result of the maximum likelihood value in the presence of this data is expected to be close to that of the maximum possible probability.

In incremental clustering, error at an earlier stage can have an impact on decisions at a later stage. Knack imposes a strict criterion for assigning new data to existing clusters in the incremental clustering phase and compensates for this by merging in the second phase. During merging, Knack uses a metric to determine the most similar cluster to a target cluster. For this metric, (chi-square) may be useful. However, our experiments with chi-square did not yield satisfactory outcomes. Knack, therefore, averages the distribution difference, i.e,

$$[p(F_i \mid C_i) - p(F_i \mid C_j)]^2 \times p(F_i \mid C_i) \qquad (5)$$

over clusters other than i. C.p.d., as discussed earlier, is constructed using a Dirichlet distribution. For a new cluster, the c.p.d. is initially assumed to be equally distributed, i.e., $1/n_f$, where n_f stands for the number of values that the i^{th} attribute possesses.

The criterion for determining the optimal number of new clusters (i.e., number of indices) is determined by the conditional probability of the number of clusters j given stored data (also see [Cheeseman etc., 1988]). At each mergence of clusters, the number of total clusters changes. It can be shown that the conditional probability is evaluated in terms of the expression:

$$(\frac{(n_f - 1)!}{(n_f + n_c + 1)!}) \times \prod n_{fc}! \qquad (6)$$

This expression will be evaluated over all attributes. The discussion is given in [Cooper & Herskovits, 1992]. The optimal number of clusters corresponds to the partition that maximizes the above expression [Chang etc., 1995]. In our experiments with categorical test data, this optimality criterion gives a reasonable prediction as we compare the results with original labels. This criterion helps to determine the number of new indices generated.

In Knack an alternative incremental clustering approach has been investigated. This approach adjusts the content of clusters for each arriving data. Once a new data arrives, Knack updates the average nearest neighbor distance, s, decides the label for the new arrival and revises (e.g., deletion, insertion) existing clusters according to s. Finally, the new arrival is classified to an existing cluster if its first nearest neighbor distance is smaller than s, otherwise, it is assigned a new label. A data point is removed from a cluster if it is at least s distance away from other points in the presence of the new arrival. This revision strategy is similar to adaptive k-means as discussed in [Jain & Dubes, 1988][Fukunaga, 1990]. Incremental clustering in COBWEB [Fisher, 1987] and Anderson's categorizing approach [Anderson, 1990] do not provide revision. The system, CLASSIT [Gennari etc., 1989], mentioned the use of operators such as split and merging for revision. The difference is that revision in Knack is based on global measurements and is used to reassign those points removed from their original clusters. Revision may take longer time but yields less misplaced

instances. Our experiments with IRIS data showed that incremental clustering with adjustment limits misclassified data to 4. As compared to other clustering systems (e.g., AUTOCLASS), Knack aims at an integrated approach. In Knack, several different clustering approaches are incorporated for comparison study.

Organization with different perspectives

The set of attributes may reflect various perspectives from which information was selected. In clustering, using too many dependent attributes may result in a pattern of ambiguity. There are two reasons: First, the measure of similarity is perspective-sensitive. For instance, Guinea pig is closer to Leopard than Macaw from the animal taxonomic perspective whereas it is closer to Macaw than Leopard if our perspective is pet. Clustering should be carried out with respect to a particular perspective. Second, if the number of attributes with one perspective significantly exceeds that of another one, the result of clustering will likely be dominated by the one with more attributes. Hence, we need separate attributes and organize cases with respect to different perspectives.

In our approach, the property of perspective is similar to the notion of a factor or a hidden structure, i.e., a subset of closely related attributes become statistically independent once the hidden structure is known. The desired hidden structure corresponds to a partition of cases. We apply clustering methods to the derivation of its property. We assume that attributes belonging to the same perspective are dependent.

The outcome may contain several subsets of attributes. Those subsets may or may not be exclusive. To show the result, we borrow the example of animal categorization from [Martin & Billman, 1994]. In this data set, information about animals are listed. Each animal is described by a list of attributes, i.e., found, food, covering, legs, mobility, reproduction and appearance. After dependency analysis the subset of attributes {found, food} as well as {covering, legs, mobility} were selected. The first subset generated two clusters and the second one generates three clusters. With externally provided information, the first subset can be identified with the notion of "affinity" with two values $< pet, wild >$ and the second one corresponds to "species" with three values $< bird, fish, mammal >$. The notions of affinity and species correspond to two different perspectives. From the perspective of species, attributes {covering, legs, mobility} are thematic while other attributes are contextual. In a probabilistic network model, a perspective is represented as a top node. In this example, the top nodes include "affinity", "species", "reproduction" and "appearance". "Covering", "legs" and "mobility" are the child nodes of "species" and "found" and "food" are attached to "affinity". The top nodes may share child nodes. With the derived information for perspectives, the joint probability density functions associated with links connecting the parent and child nodes can be determined. By adjusting the clustering criteria, we could obtain different taxonomical structures with multiple layers.

Summary

As we have discussed, the Knack system was designed as an experimental tool supporting basic adaptive retrieval and indexing functions. Important functions for computing weights involve weighting and selecting attributes, learning dependency relationships and calculating typicality for each stored case. Informativeness measures are used to guide the search. In dynamic environments, the informativeness of attributes varies. Hence its value is updated in response to the environmental change. For non-parametric estimation, new cases are simply stored away, while for parametric density functions, the joint conditional probability density is calculated assuming independent occurrence of cases. In Knack, the computation of similarity is viewed as evidence combination, rather than simple vector operations. As we just mentioned, two influential factors used for evidence combination are the informativeness and dependency relationships of attributes.

New indices for case organization are generated using clustering methods. Knack integrates properties of incremental and hierarchical clustering. New indices may be in the form of new terms or in the form of probability distributions. The case structure can be a probabilistic network model with new labels as the parent node and attributes as the child nodes or could be a discriminant network with one clustering outcome as the parent node and its next generation of clustering outcomes as the child nodes. Case retrieval with new indices is expected to be more effective and coherent. We mentioned that information about perspective needs to be incorporated into case organization. The perspective information was represented in terms of hidden structure and the result suggested that cases should be organized in multiple ways. As a result, different cases could be retrieved from different perspectives.

Knack facilitates different response modes. The user may choose automatic clustering which requires the least intervention or the try-and-error (manual) mode where the user is asked to provide information at each step. The discussion here does not imply that the functions provided in Knack are inclusive. They can be considered a typical starting set.

Appendix: Describing the contribution

A. Reasoning framework:

Knack is a testbed that integrates a number of tools for managing classification and clustering as it pertains to CBR. It assumes the data is in the form of feature/value matrices. In Knack, the notion of adaptation is realized by dynamically updating similarity measurements and re-indexing stored cases. Re-

indexing with prominent attributes assists the control of retrieval. Knack provides facilities for data management, for creating and saving model configurations to be used in experiments so that they can be used in subsequent experiments. It provides means for experimenting with different model configurations.

B. Task:

The task is to provide a testbed for experimenting with memory models used in classification and clustering. The testbed provides a basis for defining models of memory and testing their utility. The tool is not domain specific but does rely on the data having been organized as a flat file. Inputs are complex including a detailed problem configuration with various weighting schemes, precision, normalization schemes and similarity functions. The tool has been designed so that the researcher can extend it with little effort using C and xwindows programming. It allows the researcher to experiment and compare various classification, clustering and combined approaches.

C. Evaluation:

Knack provides a number of quantitative evaluation metrics and tools for testing and evaluating models. Evaluation metrics include those associated with various aspects of classification and clustering techniques and well as those for multiple model evaluation in terms of configurations and performance. Knack IS a testbed and is primary an evaluation instrument!

Acknowledgments

The authors would like to thank David Aha for his assistance in the preparation of this paper and two anonymous reviewers for helpful suggestions.

References

Aha, D. W., & Bankert, R. L. (1994). Feature selection for case-based classification of cloud types. In *AAAI Workshop on Case-Based Reasoning*, AAAI Press, 106–112.

Anderson, J. (1990). *The adaptive character of thought.* Hillsdale, NJ: Erlbaum.

Berger, J. (1985). *Statistical decision theory and Bayesian analysis.* New York: Springer-Verlag.

Chang, L., Harrison, P. & Davis, L. (1995). Report on experiments and implementation of Knack. NRL TR.

Cheeseman, P., Self, M., Taylor, W. & Freeman, D. (1988). Autoclass: a Bayesian classification system. In *Proceedings of the Fifth International Conference on Machine Learning*, 54–64.

Cooper, G. & Herskovits, E. (1992). A Bayesian Method for the Induction of Probabilistic Networks from Data In *Machine Learning, 9*, 309–347.

Dempster, A., Laird, N. & Rubin, D. (1989). Maximum likelihood from incomplete data via the EM algorithm, *J. Roy. Statist. Soc. B, 39*, (pp. 1-38).

Fisher, D. (1987). Conceptual clustering, learning from examples, and inferences. *Proceedings of the fourth International Workshop on Machine Learning* (pp. 38–49). Irvine, CA: Morgan Kaufmann.

Fukunaga, K. (1990). *Statistical Pattern Recognition.* Academic Press, INC.

Gennari, J., Langley, P. & Fisher, D. (1989). Models of Incremental Concept Formation, *Artificial Intelligence, 40*(1-3), (pp. 11-61).

Harrison, P., Chang, L. & Meyrowitz, A. (1994). Memory organization for case-based reasoning, *Proceedings of World Congress on Expert Systems.*

Jain, A. & Dubes, R. (1988). *Algorithms for clustering data.* Prentice Hall, Englewood Cliffs, NJ.

Kolodner, J. (1993). *Case-based reasoning.* San Mateo, CA: Morgan Kaufmann.

Langley, P. (1994). Selection of relevant features in machine learning (Technical Report 94-3). Stanford, CA: Institute for the Study of Learning and Expertise.

Martin, J. & Billman, D. (1994). Acquiring and combining overlapping concepts. *Machine Learning, 16*, 121–155.

Musman, S., Chang, L. & Booker, L. (1991). Application of a real-time control strategy for Bayesian belief networks to ship classification problem solving. *International Journal of Pattern Recognition and Artificial Intelligence, 7*, 513–526.

Pazzani, M. (1995). Searching for attribute dependencies in Bayesian classifiers. *Proceedings of the Fifth Workshop on Artificial Intelligence and Statistics* (pp. 424–429). Ft Lauderdale, FL: Unpublished.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: Networks of plausible inference.* San Mateo, CA: Morgan Kaufmann.

Quinlan, J. (1992). *C4.5.* San Mateo, CA: Morgan Kaufmann.

Ram, A. & Santamaria, J. (1993). Multistrategy Learning in Reactive Control Systems for Autonomous Robotic Navigation. *Informatica, 17*, 347–369.

Schank, R. (1982). *Dynamic memory: A theory of learning in computers and people.* New York: Cambridge University Press.

Stanfill, C. & Waltz, D. (1986). Toward memory-based reasoning. *Communications of the ACM, 29*(12), 1213–1228.

Verma, T. & Pearl, J. (1990). Equivalence and synthesis of causal models. *Proceedings of the Conference on Uncertainty in AI* (pp. 220–227).

Wettschereck, D. & Dietterich, T. (1995). An experimental comparison of the nearest neighbor and nearest hyperrectangle algorithms. *Machine Learning, 19*, 5–28.