Case-Based Learning: Beyond Classification of Feature Vectors*

David W. Aha[1] and Dietrich Wettschereck[2]

[1] Navy Center for Applied Research in Artificial Intelligence, Naval Research Laboratory, Washington, DC USA, aha@aic.nrl.navy.mil
[2] FIT.KI, GMD (German National Research Center for Information Technology), Schloß Birlinghoven, 53754 Sankt Augustin, Germany, dietrich.wettschereck@gmd.de

Abstract. The dominant theme of case-based research at recent ML conferences has been on classifying cases represented by feature vectors. However, other useful tasks can be targeted, and other representations are often preferable. We review the recent literature on case-based learning, focusing on alternative performance tasks and more expressive case representations. We also highlight topics in need of additional research.

1 Introduction

The majority of machine learning (ML) research has focussed on *supervised learning* tasks in which class-labeled cases, each represented as a vector of *features*, are given to a learning algorithm that induces a *concept description*. This description can then be used to predict the class labels of unlabeled cases. One approach for solving supervised learning tasks, called *case-based*,[3] involves storing cases, often as ⟨problem,solution⟩ pairs, and retrieving them to solve similar problems. This distinguishes their behavior from approaches that greedily replace cases with abstract data structures (e.g., decision trees, rule sets, artificial neural networks, and Bayesian nets).

ML research on case-based approaches frequently focuses on classifying feature vectors (e.g., Aha et al., 1991; Cost & Salzberg, 1993; Wettschereck & Dietterich, 1995). This is not surprising given that several classes of related algorithms also focus on classifying feature vectors, including *k-nearest neighbor classifiers*, *locally weighted learners*, *radial basis function networks*, and *exemplar-based models* of human concept formation. However, this restriction tends to slow scientific progress on other case-based learning (CBL) issues.

We advocate greater emphasis on alternative performance tasks and case representations, where several significant problems remain unsolved. Section 2 reviews the case-based reasoning (CBR) problem solving cycle, identifies several of its learning issues, and reviews related research. Because this cycle is rife with other interesting research topics, few CBR researchers focus on developing learning algorithms, and ML researchers may be unfamiliar with some of its

* ECML-97 invited paper, AIC-97-002. This accompanies the ECML-97 MLNet Workshop of the same title (http://www.aic.nrl.navy.mil/~aha/ecml97-wkshp).

[3] Popular synonyms include *instance-based*, *memory-based*, and *exemplar-based*.

other learning opportunities. Therefore, we outline some open CBL research issues in Section 3. Space constraints prevent us from reviewing CBL research on classifying feature vectors here, which is the focus of (Aha, 1997a).

2 Case representation and case-based learning

We begin by describing the CBR problem solving cycle and its knowledge sources, and then use them to organize our brief description of relevant CBL research. The CBR problem solving cycle (Aamodt & Plaza, 1994) contains four steps:

1. *Retrieval*: Retrieve a set of cases whose problems are similar to the query.
2. *Reuse*: Apply the retrieved cases' solutions to solve the query.
3. *Revision*: Evaluate this solution, and revise it as necessary.
4. *Retention*: Add the new ⟨query problem,solution⟩ pair to the case library.

Most CBL research on classifying feature vectors has focussed on modifying retrieval knowledge (e.g., new similarity functions, fast indexing strategies). Other issues concerning retrieval (e.g., modifying retrieval behavior by applying ease-of-revision constraints) are usually not reported at ML conferences. This also applies to the other steps in this cycle, primarily because solutions in classification tasks are class labels, which do not require complex reuse and revision strategies. However, solutions in cases can be plans, designs, or other complex structures, which complicates case reuse and revision.

According to Richter (1995), four *containers* of knowledge interact with these steps in the CBR cycle:

1. the case description language,
2. the similarity measure (for case retrieval),
3. the solution transformation (for adaptation during case revision), and
4. the cases themselves.

The following subsections summarize CBL research that targets one or more of these containers. We focus on research that either uses alternatives to feature vector representations for cases and/or addresses tasks other than classification.

2.1 Alternative representations for describing cases

Several representations for cases have been investigated (Gebhardt, 1996). For example, cases in case-based design systems (Maher et al., 1996; Börner, 1995; Surma & Braunschweig, 1996) are complex design artifacts. These require structural similarity functions, and the performance task is typically design construction rather than classification.

Other performance tasks also typically influence choices for representing cases. For example, CIBL (Branting & Broos, 1997) learns the concept of a *preference pair*, where each case is an ordered pair of states (feature vectors) in which the first is preferable to the second. Given a new pair of states, it predicts which is preferable. Langley and Pfleger (1995) instead represented cases as *evidence grids* (i.e., two-dimensional matrices, where each element is a probability estimate that its location contains a tangible object). Their algorithm learns to classify a given location as one of a set of known places.

Sequential problem-solving tasks, such as two person games, often demand complex case representations. Kerner's (1995) chess player represents cases as frames that serve as *explanation patterns* to evaluate game patterns. He described several learning operators for modifying them. CABOT (Callan et al., 1991), an Othello player, represents cases as ⟨state,move⟩ pairs. Given a query, it retrieves a most similar (game) state, and then uses a second similarity computation to select the move that best mimics the retrieved case's implicit state changes. If an oracle indicates a selection error occurred, then CABOT updates one of its two similarity functions or stores a new case. Finally, Elliot and Scott's (1991) algorithm solves integration problems, where cases (expressions) are nodes in a tree-structured hierarchy and similarity is a function of hierarchy distance. Learning involves modifying counts indicating whether operators were successfully applied, and associating an operator for each expression.

Horn clauses have also been used to represent cases. Grolimund and Ganascia (1995) used them for an optimization task. Their learning algorithm stores and reuses operator selection experience during a tabu search. RIBL (Emde & Wettschereck, 1996), a first-order classifier, instead constructs cases from multirelational data and computes the similarities between arbitrarily complex cases. Finally, Malek and Rialle (1994) described a medical diagnosis task and learning procedures for modifying Horn clause cases.

Other case representations that have been used include graphs, which require attention to the subgraph isomorphism problem. Messmer and Bunke (1995) described a TDIDT approach that allows graph-structured cases to be retrieved in polynomial time, but it trades off exponential space. Applying tree simplification strategies (e.g., pruning) can decrease space requirements, but they also reduce gains in computational efficiency. In contrast, cases in CABINS (Miyashita & Sycara, 1993) are schedules; it improves their quality using an iterative refinement process. Ramsey and Grefenstette (1994) defined a case as a (genetically induced) rule population. Their CBL algorithm uses environmental cues to select which population to use in response to concept drifts. Finally, *case retrieval nets* represent cases as sets of entities in a joint semantic network. Lenz (1996) described their application to classification tasks.

2.2 Learning retrieval knowledge

Complex representations and tasks (e.g., planning) require learning algorithms tailored to their case retrieval and revision needs. For example, CAPLAN/CBC (Muñoz-Avila & Hüllen, 1996) learns feature weights to index plan descriptions, and RUNNER (Seifert et al., 1994) learns operator application conditions and uses them to index cases represented by semantic networks. HAMLET (Borrajo & Veloso, 1997) learns to improve its search efficiency and resulting plan quality by incrementally refining its control rules using a CBL approach. META-AQUA

(Cox & Ram, 1994) uses explicit learning goals to select strategies for recovering from planning failures. Krovvidy and Wee's (1993) CBL system saves partial planning solutions and learns heuristics for selecting them during problem solving. Knowledge on the adaptability of retrieved cases can also be used to successfully bias retrieval behavior (Smyth & Keane, 1995a).

2.3 Learning adaptation knowledge

Learned adaptation knowledge can be used to bias the case vocabulary. For example, ROBBIE (Fox & Leake, 1995) uses constructive induction to learn new indices by estimating their ability to retrieve more easily adapted cases. Adaptation knowledge can also be learned by integrating case- and rule-based approaches (Leake et al., 1996), by evolutionary algorithms (Hunt, 1995), or by analyzing case comparisons (Hanney & Keane, 1996). CARMA (Hastings et al., 1995) instead learns weights on adaptation operators.

Approaches that annotate case solutions with explicit justification knowledge for guiding reuse and revision employ a form of adaptation known as *derivational analogy*. This is most often used in planning tasks. Veloso and Carbonell (1993) described a derivational approach that learns to improve case indices and speed problem solving. Bhansali and Harandi (1993) used derivational analogy to learn how to synthesize UNIX programs. In both cases, several learning opportunities arose, including how to use justification knowledge to direct the search for selecting goals, operators, and bindings.

2.4 Learning and case libraries

The final knowledge container is the case library itself. Because new cases do not always contribute sufficient information to warrant their retention, several case-based systems selectively retain cases. For example, Smyth and Keane (1995b) described algorithms that delete cases whose solutions are covered by other cases. Aha (1997b) instead proposed how learning algorithms can be used to modify cases rather than delete them.

3 Some suggested research topics

Several CBL research issues require attention, both in the context of specific performance tasks and as components in integrated learning frameworks. The following subsections briefly describe some of these issues.

3.1 Data mining

Data mining is a step in the *knowledge discovery process* in which inferences are generated from large databases (Fayyad et al., 1996). Case-based learning systems can assist in data mining activities in several ways. For example, CBL can be used to locate extreme cases (i.e., *gems*) that are of particular interest. Also, CBL is a promising approach for mining in multi-relational databases (Emde & Wettschereck, 1996), which has not yet been extensively investigated even though many large databases employ relational representations. Data mining requires attention to efficiency concerns, which could spur research on designing typicality-guided retrieval strategies (Porter et al., 1990), parallel case retrieval (Kettler et al., 1994), and software support for case construction (Kitano et al., 1993). Also needed are techniques for validating the behavior of case revision operators, and further research on how abstraction can be used to reduce search without requiring that solutions be downward refinable (Branting & Aha, 1995).

3.2 Synthesis performance tasks

Most CBL research has focussed on analysis (e.g., classification) rather than synthesis (e.g., design, planning) tasks. It is not clear what modifications of techniques for classifying feature vectors are needed for synthesis tasks. For example, for problems described by feature collections, Muñoz-Avila and Hüllen (1996) showed how feature weighting can be used in planning tasks, but it is unclear how weighting algorithms should be designed for problems represented by graphs. Similarly, adaptation has a more complex role in synthesis than in most classification tasks. More robust techniques are needed for learning and modifying constraints in case-based planning and design systems (Avesani et al., 1993; Purvis & Pu, 1995).

Several case-based systems, especially those that target legal analysis, learn to synthesize *explanatory arguments* for (and sometimes also against) specific decisions (e.g., Branting, 1990). Additional CBL research is needed on inducing rules to supplement case-based argumentation, such as by identifying case retrieval contexts for given legal situations, and by updating indexing topologies to ensure that the retrieved case solutions are appropriate.

3.3 Components in integrated learning frameworks

CBL algorithms have been integrated with learning algorithms that target a variety of performance tasks, including knowledge acquisition (Tecuci, 1993), analytic problem solving (Borrajo & Veloso, 1997), reinforcement learning (McCallum, 1995), and Bayesian reasoning (Tirri et al., 1996). These integrations frequently use CBL in innovative fashions. For example, McCallum's cases provide historical information that allow agents to distinguish perceptually identical locations, while Borrajo and Veloso used several lazy explanation-based learning techniques to improve plan quality and reduce planning time. However, these integrations often introduce several novel research issues. For example, Tecuci showed how simple determinations can help induce structured explanations for observations, but leaves open the question of whether and how more elaborate CBL techniques (e.g., derivational analogy) can be used to dynamically formulate cases and acquire knowledge. Similarly, additional research is needed to investigate how CBL techniques can assist in inducing Bayes networks, where cases are individual networks. Many other opportunities for novel research contributions exist in the context of integrating CBL with other learning approaches.

4 Conclusion

Previous machine learning research on case-based approaches has typically focussed on classifying feature vectors. Although much progress has been made, attention to several other case-based learning issues is needed. This paper briefly outlines existing research on case-base learning that goes beyond classifying feature vectors, and suggests directions for future research.

Acknowledgements

Thanks to Len Breslow, Héctor Muñoz-Avila, Jerzy Surma, and Henry Tirri for their comments on an earlier draft of this paper. This research was supported by the Office of Naval Research.

References

Aamodt, A., & Plaza, E. (1994). Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Communications*, *7*, 39–59.

Aha, D. W. (1997a). A brief history of case-based classification. Unpublished manuscript. http://www.aic.nrl.navy.mil/~aha.

Aha, D. W. (1997b). A proposal for refining case libraries. To appear in *Proceedings of the Fifth German Workshop on Case-Based Reasoning.*

Aha, D. W., Kibler, D., & Albert, M. K. (1991). Instance-based learning algorithms. *Machine Learning*, *6*, 37–66.

Avesani, P., Perini, A., & Ricci, F. (1993). Combining CBR and constraint reasoning in planning forest fire fighting. *Proceedings of the First European Workshop on Case-Based Reasoning* (pp. 235–239). Kaiserslautern, Germany: Unpublished.

Bhansali, S., & Harandi, M. T. (1993). Synthesis of UNIX programs using derivational analogy. *Machine Learning*, *10*, 7–55.

Borrajo, D., & Veloso, M. (1997). Lazy incremental learning of control knowledge for efficiently obtaining quality plans. To appear in *Artificial Intelligence Review.*

Börner, K. (Ed.) (1995). *Modules for design support* (Technical Report 35). Freiburg, Germany: University of Freiburg, Center for Cognitive Science.

Branting, L. K. (1990). *Integrating rules and precedents for classification and explanation: Automating legal analysis* (Technical Report AI90-146). Austin, TX: University of Texas, Artificial Intelligence Laboratory.

Branting, L. K., & Aha, D. W. (1995). Stratified case-based reasoning: Reusing hierarchical problem solving episodes. *Proceedings of the 14th International Joint Conference on Artificial Intelligence* (pp. 384–390). Montreal: Morgan Kaufmann.

Branting, L. K., & Broos, P. (1997). Automated acquisition of user preferences. *International Journal of Human-Computer Studies, 46*, 55–77.

Callan, J. P., Fawcett, T. E., & Rissland, E. L. (1991). CABOT: An adaptive approach to case-based search. *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence* (pp. 803–808). Sydney: Morgan Kaufmann.

Cost, S., & Salzberg, S. (1993). A weighted nearest neighbor algorithm for learning with symbolic features. *Machine Learning*, *10*, 57–78.

Cox, M., & Ram, A. (1994). Managing learning goals in strategy-selection problems. In *Working Papers of the Second European Workshop on Case-Based Reasoning* (pp. 85–93). Chantilly, France: Unpublished.

Elliot, T., & Scott, P. D. (1991). Instance-based and generalization-based learning procedures applied to solving integration problems. *Proceedings of the 8th Conference of the Society for the Study of AI* (pp. 256–265). Leeds, UK: Springer.

Emde, W., & Wettschereck, D. (1996). Relational instance-based learning. *Proceedings of the Thirteenth International Conference on Machine Learning* (pp. 122–130). Bari, Italy: Morgan Kaufman.

Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., & Uthurusamy, R. (Eds.), *Advances in Knowledge Discovery and Data Mining.* Cambridge, Massachusetts: MIT Press.

Fox, S., & Leake, D. L. (1995). Using introspective reasoning to refine indexing. *Proceedings of the 14th International Joint Conference on Artificial Intelligence* (pp. 391–397). Montreal: Morgan Kaufmann.

Gebhardt, F. (1996). Structure oriented case retrieval. *Fourth German Workshop on Case-Based Reasoning: System Development and Evaluation* (pp. 95–102). Technical Report, Humbold University, Informatik-Bericht Number 55.

Grolimund, S., & Ganascia, J.-L. (1995). Integrating case-based reasoning and tabu search for solving optimisation problems. *Proceedings of the First International Conference on Case-Based Reasoning* (pp. 451–460). Sesimbra, Portugal: Springer.

Hanney, K., & Keane, M. T. (1996). Learning adaptation rules from a case-base. *Proceedings of the 3rd European Workshop on CBR* (pp. 179–192). Lausanne: Springer.

Hastings, J., D., Branting, L. K., & Lockwood, J. A. (1995). Case adaptation using an incomplete causal model. *Proceedings of the First International Conference on Case-Based Reasoning* (pp. 181–192). Sesimbra, Portugal: Springer.

Hunt, J. (1995). Evolutionary case based design. In I. D. Watson (Ed.), *Progress in Case-Based Reasoning: First United Kingdom Workshop.* Salford, UK: Springer.

Kerner, Y. (1995). Learning strategies for explanation patterns: Basic game patterns with application to chess. In *Proceedings of the First International Conference on Case-Based Reasoning* (pp. 491–500). Sesimbra, Portugal: Springer.

Kettler, B. P., Hendler, J. A., Andersen, W. A., & Evett, M. P. (1994). Massively parallel support for case-based planning. *IEEE Expert*, *2*, 8–14.

Kitano H., Shimazu H., & Shibata A. (1993). Case-method: A methodology for building large-scale case-based systems. *Proceedings of the Eleventh National Conference on Artificial Intelligence* (pp. 303–308). Washington, DC: AAAI Press.

Krovvidy, S., & Wee, W. G. (1993). Wastewater treatment systems from case-based reasoning. *Machine Learning*, *10*, 341–363.

Langley, P., & Pfleger, K. (1995). Case-based acquisition of place knowledge. *Proceedings of the Twelfth International Conference on Machine Learning* (pp. 344–352). Tahoe City, CA: Morgan Kaufmann.

Leake, D. B., Kinley, A., & Wilson, D. (1996). Acquiring case adaptation knowledge: A hybrid approach. *Proceedings of the Thirteenth National Conference on Artificial Intelligence* (pp. 684–689). Portland, OR: AAAI Press.

Lenz, M. (1996). Case retrieval nets applied to large case bases. *Fourth German Workshop on Case-Based Reasoning: System Development and Evaluation* (pp. 111–118). Technical Report, Humbold University, Informatik-Bericht Number 55.

Maher, M. L., Balachandran, M. B., & Zhang, D. M. (1995). *Case-Based Reasoning in Design.* Mahwah, NJ: Erlbaum.

Malek, M., & Rialle, V. (1994). A case-based reasoning system applied to neuropathy diagnosis. *Proceedings of the Second European Workshop on Case-Based Reasoning* (pp. 329–336). Chantilly, France: Unpublished.

McCallum, R. A. (1995). Instance-based utile distinctions for reinforcement learning. *Proceedings of the Twelfth International Conference on Machine Learning*. Lake Tahoe, CA: Morgan Kaufmann.

Messmer, B. T., & Bunke, H. (1995). *Subgraph isomorphism in polynomial time* (Technical Report IAM 95-003). Bern, Switzerland: University of Bern, Institute of Computer Science and Applied Mathematics.

Miyashita, K., & Sycara, K. (1993). Improving schedule quality through case-based reasoning. In D. Leake (Ed.), *Case-Based Reasoning: Papers from the 1993 Workshop* (Technical Report WS-93-01). Menlo Park, CA: AAAI Press.

Muñoz-Avila, H., & Hüllen, J. (1996). Feature weighting by explaining case-based problem solving episodes. *Proceedings of the Third European Workshop on Case-Based Reasoning* (pp. 280–294). Lausanne, Switzerland: Springer.

Porter, B. W., Bareiss, R., & Holte, R. C. (1990). Knowledge acquisition and heuristic classification in weak-theory domains. *Artificial Intelligence*, *45*, 229–263.

Purvis, L. & Pu, P. (1995). Adaptation using constraint satisfaction techniques. *Proceedings of the First International Conference on Case Based Reasoning* (pp. 289–300). Sesimbra, Portugal: Springer Verlag.

Ramsey, C. L., & Grefenstette, J. J. (1994). Case-based anytime learning. In D. W. Aha (Ed.), *Case-Based Reasoning: Papers from the 1994 Workshop* (Technical Report WS-94-01). Menlo Park, CA: AAAI Press.

Richter, M. (1995). The knowledge contained in similarity measures. http://wwwagr.informatik.uni-kl.de/ lsa/CBR/Richtericcbr95remarks.html.

Seifert, C. M., Hammond, K. J., Johnson, H. M., Converse, T. M., McDougal, T. F., & Vanderstoep, S. W. (1994). Case-based learning: Predictive features in indexing. *Machine Learning*, *16*, 37–56.

Smyth, B. & Keane, M. T. (1995a). Experiments on adaptation-guided retrieval in case-based design. *Proceedings of the First International Conference on Case-Based Reasoning* (pp. 313–324). Sesimbra, Portugal: Springer.

Smyth, B. & Keane, M. T. (1995b). Remembering to forget: A competence-preserving deletion policy for case-based systems. *Proceedings of the 14th International Joint Conference on Artificial Intelligence* (pp. 377–383). Montreal: Morgan Kaufmann.

Surma, J., & Braunschweig, B. (1996). REPRO: Supporting flowsheet design by case-based retrieval. *Proceedings of the Third European Workshop on Case-Based Reasoning* (pp. 400–412). Lausanne, Switzerland: Springer.

Tecuci G., (1993). Plausible justification trees: A framework for deep and dynamic integration of learning strategies. *Machine Learning, 11*, 237–261.

Tirri, H., Kontkanen, P., & Myllmäki, P. (1996). A Bayesian framework for case-based reasoning. *Proceedings of the Third European Workshop on Case-Based Reasoning* (pp. 413–427). Lausanne, Switzerland: Springer.

Veloso, M. M., & Carbonell, J. G. (1993). Derivational analogy in PRODIGY: Automating case acquisition, storage, and utilization. *Machine Learning*, *10*, 249–278.

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

This article was processed using the LaTeX macro package with LLNCS style