Remembering Why to Remember: Performance-Guided Case-Base Maintenance*

David B. Leake and David C. Wilson

Computer Science Department
Lindley Hall 215, Indiana University
Bloomington, IN 47405, U.S.A.
{leake,davwils}@cs.indiana.edu

Abstract. An important focus of recent CBR research is on how to develop strategies for achieving compact, competent case-bases, as a way to improve the performance of CBR systems. However, compactness and competence are not always good predictors of performance, especially when problem distributions are non-uniform. Consequently, this paper argues for developing methods that tie case-base maintenance more directly to performance concerns. The paper begins by examining the relationship between competence and performance, discussing the goals and constraints that should guide addition and deletion of cases. It next illustrates the importance of augmenting competence-based criteria with quantitative performance-based considerations, and proposes a strategy for closely reflecting adaptation performance effects when compressing a case-base. It then presents empirical studies examining the performance tradeoffs of current methods and the benefits of applying fine-grained performance-based criteria to case-base compression, showing that performance-based methods may be especially important for task domains with non-uniform problem distributions.

1 Introduction

Case-base maintenance has become an active CBR research area, producing results with important ramifications for both the theory and practice of CBR. Much significant work in this area focuses on developing methods for reducing the size of the case-base while maintaining *case-base competence*, "the range of target problems that can be successfully solved" (Smyth & McKenna 1999a). Strategies have been developed for controlling case-base growth through methods such as competence-preserving deletion (Smyth & Keane 1995) and failure-driven deletion (Portinale, Torasso, & Tavano 1999), as well as for generating compact case-bases through competence-based case addition (Smyth & McKenna 1999a; Zhu & Yang 1999).

* The authors' research is supported in part by NASA under award No NCC 2-1035. Copyright ©2000 Springer–Verlag. This paper appears in the *Proceedings* of EWCBR-2K but is not camera-identical to the proceedings version. This paper is available from the archive at `http://www.cs.indiana.edu/~leake/INDEX.html`.

The goal of achieving compact competent case-bases addresses important performance objectives for CBR systems. First, sufficient competence is a *sine qua non* for performance: no CBR system is useful unless it can solve a sufficient proportion of the problems that it confronts. Second, compacting the case-base may help to increase system efficiency by alleviating the utility problem for retrieval (Francis & Ram 1993; Smyth & Cunningham 1996). As an added benefit, compact case-bases decrease communications costs when case-bases are used as vehicles for knowledge sharing or are transferred in distributed CBR systems (cf. (Doyle & Cunningham 1999)).

However, case-base compactness is only a proxy for performance in a CBR system, rather than an end in itself. For example, decreased retrieval cost from a smaller case-base may be counterbalanced by increased adaptation costs or decreased quality. Thus optimizing the performance of a CBR system may require balancing tradeoffs between competence, quality, and efficiency (Portinale, Torasso, & Tavano 1999; Smyth & Cunningham 1996). In addition, adjusting the case-base to optimize performance may require reasoning about the system's performance environment, taking into account that patterns in problem distribution make some cases more useful than others (Leake & Wilson 1999). Consequently, effective maintenance requires remembering *why* cases are being remembered (or forgotten)—to serve the overall performance goals of the CBR system for a given task—and optimizing maintenance decisions accordingly. Now that research on case-base competence is becoming mature, we believe that the time is ripe to make performance criteria play a more direct role in guiding case addition and deletion.

This paper examines the benefits of using fine-grained performance metrics to directly guide case addition and deletion, and presents initial experiments on their practicality. The paper begins by discussing the competence/performance dichotomy and the factors that should guide case-base maintenance. It then illustrates the importance of adding direct performance considerations to maintenance strategies, by showing that in some cases, increased performance can be achieved without sacrificing either competence or compactness. It next presents a performance-based metric, guided by cases' contributions to adaptation performance, to guide case addition and deletion. Experiments examine the common alternative practice of reflecting performance with fixed adaptation effort thresholds, illuminating tradeoffs in adaptation cost and case-base compression, and then compare the effects of competence-based and performance-based strategies. Our results show that performance-based deletion strategies are especially promising for non-uniform problem distributions, which have received little attention in previous analyses of case-based maintenance, but which are often important in real-world contexts.

2 The Competence-Performance Dichotomy

Case-base maintenance is fundamentally driven by performance concerns. For example, Leake and Wilson's (1998) definition of case-base maintenance is explicitly performance-related:

> Case-base maintenance implements policies for revising the contents or organization of the case-base in order to facilitate future reasoning for a particular set of performance objectives.

In this definition, the performance measure evaluates the performance of a particular CBR system for a given initial case-base and sequence of target problems.

To relate the competence and performance of CBR systems, it is useful to revisit the notions of competence and performance. When Chomsky (1965) formulated the original competence-performance dichotomy in linguistics, he used competence to describe the "in principle" abilities of an ideal speaker, unaffected by factors such as processing limitations, and used performance to describe how language was actually used by real speakers under real constraints in real situations. "Competence" in CBR has a specialized meaning—the range of target problems that a system can solve (Smyth & McKenna 1999a)—but the idea of "problems that a system *can* solve" can be taken to reflect an idealized competence. For example, if retrieval and adaptation time are allowed to be arbitrarily long, the competence of the case base for a sequence of input problems depends only on the "in principle" adequacy of system knowledge.

In practice, processing constraints are important, and current case-base competence research often reflects them in adaptation effort thresholds, which treat a case to be "adaptable" to solve a problem only if it can be adapted within a fixed limit on the number adaptation steps allowed (e.g., (Portinale, Torasso, & Tavano 1999; Zhu & Yang 1999)). Defining competence in terms of cases within the adaptation threshold combines one aspect of "idealized" competence (that the set of cases can be partitioned into adaptable and non-adaptable cases, with all adaptable cases treated as being equivalent) with the pragmatic concerns reflected in guaranteeing an upper bound on the required adaptation effort.

This paper argues for a finer-grained approach, which we call *performance-based*, to make its decisions directly reflect expected impact on top-level performance goals (in these examples, goals for processing efficiency). In order to develop this approach, we first identify the relevant performance goals and their relationships.

3 Performance Goals for Case-Base Maintenance

In general, there will be multiple performance measures for a CBR system, and there is no guarantee that all of them can be maximized simultaneously. In order to balance these measures to achieve the best overall performance, it is useful to distinguish top-level goals from goals that are only instrumental, rather than targets in themselves. For example, the goal of decreasing case-base size is not pursued for its own sake (provided space is available), but instead, as an instrumental goal of the higher-level goal to decrease retrieval time. Decreasing retrieval time is itself an instrumental goal to the top-level performance goal of improving problem-solving speed. A maintenance system that recognizes that compactness is an instrumental goal, rather than a top-level goal, can make better decisions about how to manage compactness compared to other goals, for example, by sacrificing compactness when it improves performance. However, when compactness is used as a proxy for efficiency and simply maximized, the maintenance process may miss better opportunities to maximize efficiency.

Smyth and McKenna (1999) define three types of top-level goals for CBR systems:

1. Problem-solving efficiency goals (e.g., average problem-solving time)
2. Competence goals (the range of target problems solved)
3. Solution quality goals for problems solved (e.g., the error level in solutions)

Any case addition/deletion strategy must be shaped by these goals and the acceptable tradeoffs between them. In addition, we note that addition and deletion strategies are also guided by the following constraints:

1. Case-base size limits (if any)
2. Acceptable long-term/short-term performance tradeoffs
3. The availability of secondary sources of cases
4. The expected distribution of future problems

For example, Smyth and Keane's (1995) competence-preserving deletion strategies reflect all of these constraints. Their deletion process keeps the case-base within acceptable size limits (constraint 1); their competence-guided choices are intended to minimize the loss of future coverage (constraint 2); their methods' deletion choices assume a uniform distribution of problems (constraint 3); and no other sources of cases are available for recovering deleted information (constraint 4), making preservation of competence a key concern. Other instantiations of these constraints would give rise to different strategies. For example, if short-term performance is crucial and long-term is less important, and current problems are concentrated in a small part of the case-base, it may be acceptable to sacrifice current competence and build it back through future learning. By their very nature, competence criteria aim at maximizing coverage, rather than trading off coverage and efficiency based on the expected problem distribution, but as we show later in the paper, making such tradeoffs may be useful for non-uniform problem distributions.

4 The Value of Performance-Based Criteria

Making the right decisions about cases to retain requires augmenting competence criteria with consideration of the performance effects of alternative cases. Usually this is thought of in terms of achieving a better tradeoff between competence and efficiency. However, in some situations, performance considerations can even

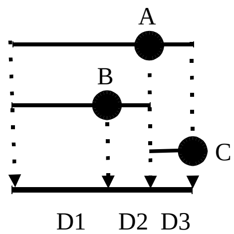

Fig. 1. Three example cases and their coverage.

improve efficiency *without loss of competence or compactness*. We illustrate this with a simple example.

For this example, we assume the most easily adaptable case is always retrieved for each problem, and that the case-base is built by from a set of candidates by a greedy algorithm which, for each step, adds the candidate case that provides the greatest increment to competence, until achieving full coverage (Zhu & Yang 1999). Consider building a case-base from 3 cases, A, B, and C as shown in Figure 1. The line segment at the bottom of the figure represents the problem space, where problems are associated with points on the line. (For example, problems could be the desired yield strength for a metal, and solutions the manufacturing processes to obtain it.) Suppose that if case C_1 solves problem p_1, the cost to adapt C_1 to solve a new problem p_2 is $\alpha|p_1 - p_2|$, for some fixed $\alpha > 0$.

The horizontal positioning of A, B, and C along the problem axis reflects the specific problems that each one solves, and the horizontal intervals adjacent to each case reflect the space of problems that it can be adapted to solve, given the system's adaptation knowledge. The interval surrounding A is an open interval on the right; case A cannot be adapted to solve the problem solved by case C. All other endpoints are closed. To build the case-base, a greedy competence-based case addition algorithm selects first case A first and then case C, resulting in the case-base $CB_1 = \{A, C\}$, which provides maximal competence. We note, that $CB_2 = \{B, C\}$ provides the same competence.

If the problem distribution is uniform, it can be shown that the difference between the expected adaptation cost for solving problems using case-base CB_1 instead of CB_2 is $\alpha D_2(D_1 - D_2/4)/(D_1 + D_2 + D_3)$. If we fix D_2 and D_3 and let $D_1 \rightarrow \infty$, the expected average adaptation cost difference goes D_2. (Intuitively, almost all problems will then fall to the left of case B, and those problems will be D_2 closer to case B than to case A.) Thus for this example, there are two competing case-bases with the same competence and the same size, but with different performance, so it is only possible to choose between them based on performance, not competence or compactness—and in fact, a competence-based greedy case addition algorithm picks the wrong one. This example demonstrates that performance-based considerations, distinct from competence and compactness, can play an important role in case-base selection.

5 A Performance-Based Metric for Case Selection

This section describes a strategy for performance-based case selection, inspired by Smyth and McKenna's (1999a) RC-CNN algorithm. That algorithm compacts case-bases using a compressed-nearest-neighbor (CNN) algorithm (Hart 1968) whose inputs are ordered by a relative coverage (RC) metric, to give priority to cases expected to make the largest competence contributions. By analogy to the RC metric, which estimates each case's unique contribution to the competence of the system, we have developed a relative performance (RP) metric aimed at assessing the contribution of a case to the *adaptation performance* of the system.

Our RP metric depends on two standard definitions from case-base competence research, the coverage set of a case (the set of problems from the target set that the problem solves) and the reachability set of a problem (the set of cases that solve that problem). It also depends on the representativeness assumption that the contents of the case-base are a good approximation of the problems the system will encounter (see (Smyth & McKenna 1999a) for full definitions and discussion), but can be weighted to reflect different expected problem frequencies.

The RP value for a case reflects how its contribution to adaptation performance compares to that of other cases. To approximate the benefit of adding the case to the case-base, we first assume that the similarity metric will accurately select the most adaptable case for any problem. For each case that might be added to the case-base, we estimate its contribution to adaptation performance. We have explored a number of metrics, including a "performance benefit" (PB) metric estimating the actual numerical savings that the addition of each case provides. However, best results were obtained by considering a case's relative adaptation performance, the percent savings it provides compared to the worst alternative case that solves the problem. If we let $RS(c', c)$ stand for $ReachabilitySet(c') - \{c\}$, for a fixed case-base CB we define:

$$RP(c) = \sum_{c' \in CoverageSet(c)} 1 - \frac{AdaptCost(c, c')}{max_{c'' \in RS(c',c)} AdaptCost(c'', c')}$$

This metric can be used to guide either case addition—favoring cases with high RP values—or case deletion—favoring cases with low RP values. By adding an additional weighting factor, reflecting the expected probability of new problems similar to those in the case-base being encountered in the input stream, this formula can reflect expected problem distributions. Even if the distribution is not known completely, this adjustment can refine case selection to improve performance for likely "hot spots" in the case-base (Leake & Wilson 1999).

Because the actual relative performance of a particular case depends on the other cases in the case-base, using completely accurate RP values to guide case deletion would require recalculating RP values after additions or deletions, which could be extremely expensive. A more practical alternative, which we will refer to as RP-CNN, is to do a one-time RP calculation, and then to use that estimate to order the cases presented to CNN, analogously to RC-CNN. A key question is whether this approximate information is sufficiently accurate to improve performance. We test RP-CNN and compare its effects to RC-CNN in Section 6.3.

6 Experimental Results

To explore the relationships between compactification strategies and performance, we conducted four experiments. These examine (1) how the choice of adaptability thresholds affects system performance, (2) the tradeoffs between compressed case-base size and expected adaptation costs for CNN, (3) the performance obtained by RC-CNN compared to RP-CNN for uniformly-distributed problems, and (4) their comparative performance for non-uniformly distributed problems.

The experiments were conducted in a simple path planning domain that models an inter-/intra-city transportation network. Concentrated areas of local connectivity represent cities. Paths are viewed as different modes of transport between locations; they do not correspond directly to grid lines but do reflect the grid distance between location points. Models are generated randomly, based on specifications of the number and size of the cities, the number of locations in each city, the minimum distance between cities, and the maximum number of paths connecting locations. The model generator ensures that all locations are reachable through some path from all other locations, if necessary adding paths to ensure connectivity.

The planner combines case-based planning with a generative (breadth-first) path planner to adapt cases by extending their paths. This enables natural control over the allowable adaptation, by setting a threshold on the allowed number of adaptation steps. Path cases represent the starting and ending locations, the path between them, and the path distance. Cases are retrieved based on minimizing the combined distance between the starting and ending locations in a case and new travel problem.

6.1 Performance Effects of Competence Coverage Thresholds

Competence-preserving addition and deletion methods must determine the competence contributions of cases, which depends on the system's ability to retrieve and adapt particular cases. As described previously, the adaptability judgment is often based on an adaptation threshold, with all cases that can be adapted within the threshold treated as equally adaptable to the problem. This blurs the adaptability differences between particular cases, sacrificing some ability to select high-performance cases. The first experiment examines affects the performance of case-bases generated by RC-CNN for different thresholds.

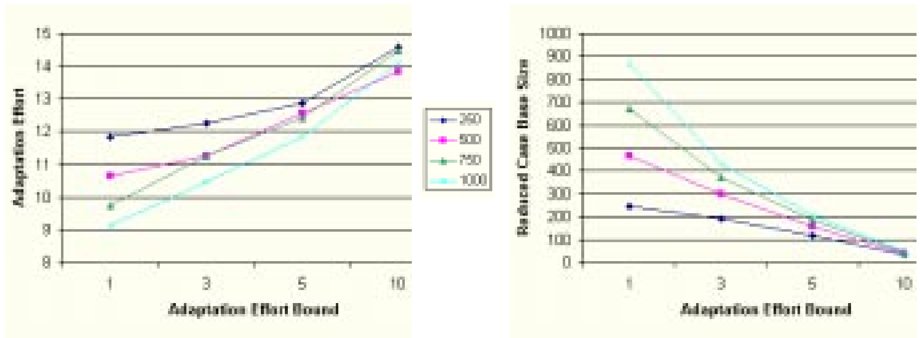

Fig. 2. Adaptation effort as a function of threshold (left), and reduced case-base size as a function of threshold (right), for RC-CNN compression.

For each test, we generated a model consisting of 3 city areas of size 20 by 20, with 40 locations in each city. We randomly generated case-bases of sizes 1000, 750, 500, and 250 from the possible starting and ending location pairs in the model. Each case-base was then reduced in size by the RC-CNN method, and the reduced case-base was tested with 100 randomly selected probes from the model space. Each test was repeated 10 times, selecting a new initial case-set and test probes for each trial and averaging the results.

Higher threshold values increase the variance in adaptation costs for problems that a case covers, decreasing pressure to add nearby cases. Consequently, we expected adaptation performance to decrease as the threshold values increased. This basic trend appears in the results in the left side of Figure 2, which shows average adaptation effort on the test problems as a function of the threshold. This effect is seen across all case-base starting sizes, but lower thresholds were better at exploiting the range of cases in large case bases, selecting closer cases (resulting in lower adaptation costs). Our explanation is that all the case-bases in our experiments were large enough to provide adequate coverage, but that, at high thresholds, case choice was not sufficiently selective to take full advantage of the wider choice of cases by choosing better case distributions.

6.2 Compressed Size vs. Adaptation Cost Tradeoffs

The previous experiment illustrates how adaptation effort thresholds used by RC-CNN can affect the adaptation effort required for a system to solve problems. However, required adaptation effort is not the only concern: There is a tradeoff because lower thresholds decrease the range of problems that each case can be used to solve, making us expect less compression to be possible for a given competence level. In this experiment we observe the effects of different case threshold levels on the case-base size obtained using RC-CNN.

Using the basic experimental procedure described previously, we determined the resulting case-base size at four different threshold levels, from 1 to 10, for initial case-bases of 250, 500, 750, and 1000 cases. We expected that as the adaptation threshold increased, the size of the case-base produced by RC-CNN would decrease. We expected that the resulting size would be ordered by the sizes of the case-bases, with the greatest compression being achieved for large case-bases. These predictions are borne out in the right side of Figure 2. It is interesting to note the very substantial compression ratio achieved for a threshold of 10.

6.3 Comparing CNN, RC-CNN and RP-CNN for Uniform Case Distributions

A third experiment compared the effects of basic CNN, RC-CNN, and RP-CNN on case-base compression and adaptation efficiency, using the same basic procedure and starting with a case-base of size 1000, with adaptation boundary of 5. For CNN, the mean case-base size was 262, for RC-CNN, 204, and for RP-CNN, 284. With a uniform distribution of test problems, mean adaptation cost for CNN was 2.96, for RC-CNN was 3.19, and for RP-CNN was 2.87. Thus as expected, RP-CNN provided some gains in efficiency at a cost of increased case-base size, while RC-CNN provided substantial gains in case-based compression at the expense of some efficiency. This provides partial independent confirmation for the results of (Smyth & McKenna 1999a).

Although RP-CNN achieved slightly better performance than the other methods, more experiments are needed (e.g., to compare the performance achieved when the size of the resulting case-bases is held constant). We have a number of refinements in mind that we expect to improve performance for the RP metric, as well as for making the RP recalculation process more efficient, in order to be able to use more accurate RP values at each step rather than relying on a single static approximation calculated at the start of processing.

6.4 RC vs. RP Deletion for Non-Uniform Case Distributions

In order to test the performance of our metric under non-uniform problem distributions, we designed an experiment in which routes with origin and destination in certain cities are requested more frequently. Both the number of cities that comprise the high traffic area and the frequency of requests for routes in that area are parameters of the experiment. At the beginning of the experiment, a subset of cities of the desired size is selected at random, and routes that start from and end in those cities are considered high-traffic routes. Test probes are randomly generated from the high-traffic areas in proportion to the specified frequency, with the remaining probes randomly generated from the lower-traffic areas.

Using the same model setup as in the earlier experiments, we tested conditions in which one and two of the three cities comprised the high-traffic routes. We ran the experiments with a 95 percent frequency rate for high-traffic probes, using the RP metric, with a weight factor to reflect the probability of a particular problem occurring (based simply on whether the problem was in a high-traffic

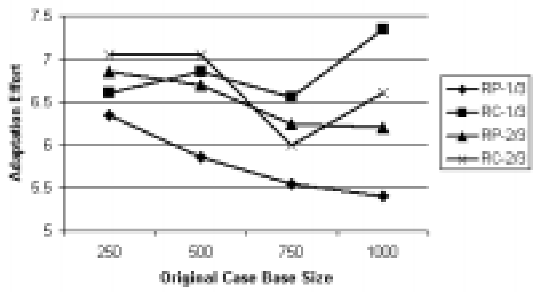

Fig. 3. Average adaptation effort for non-uniform case distributions.

area, and the probability of problems in that area). We evaluated effects on compression by case deletion, first running CNN to determine a target size for the compressed case-base, then ordering the candidate cases according to the metric being tested (RC or RP), and deleting the least desirable cases according to the metrics, until reaching the determined size. Here we expected to see greater performance benefits for RP than in the previous experiment, because RC focuses on coverage alone, while the revised RP favors useful cases in high-traffic areas. This was borne out in our results, which are shown in figure 3 for two experimental configurations: one high-traffic area and two low-traffic areas of equal size (1/3), and two high-traffic areas and one low-traffic area (2/3), for RC-CNN using an adaptation threshold of 10. The graph shows the median effort to solve cases after reduction of the case-base, for cases within the adaptation limit, for initial case-base sizes ranging from 250 to 1000 cases. For all but one test, performance with RP surpasses RC. Benefits are strongest with more focused areas (1/3), and benefits of RP appear to increase with larger initial case-bases, perhaps because the wide range of cases allows RP to fine-tune its choices.

7 Comparison to Previous Research

The importance of utility-based considerations for maintenance is well-known. Smyth and Keane's (1995) seminal competence work, for example, proposes footprint-utility deletion, in which case deletion decisions are based first on competence categories and then on utility. Smyth and Cunningham (1996) examine the tradeoffs between coverage, quality, and efficiency, illustrating how case-base size can affect retrieval and adaptation costs, as well as quality. van Someren, Surma, and Torasso (1997) suggest using on a cost model for the CBR system to guide decisions about the size of the case-base.

Portinale, Torasso and Tavano (1998) present a case deletion strategy aimed at favoring useful cases in a combined CBR-MBR system. Their method replaces old cases with new cases solved by the MBR system, provided the new case covers the problem of the replaced case, within a fixed adaptation effort threshold, and requires more effort than the case being replaced. The *Learning by Failure with Forgetting* strategy (Portinale, Torasso, & Tavano 1999) applies another heuristic, periodically deleting cases that have remained unused longer than a predefined time window and "false positive cases." These are valuable heuristic methods, but differ from the RP metric's more quantitative approach, which balances the expected future performance contributions of alternative cases in the global context of competing cases in the case-base, rather than assessing cases independently.

As discussed previously, the framework here is based on the competence modeling framework of (Smyth & McKenna 1999a). We agree with the importance of competence criteria, and plan to develop combined competence/performance metrics for tuning the maintenance process to achieve a desired balance between competence and performance concerns. For example, in a combined CBR+MBR system that can solve any problem from scratch, it may be appropriate to base maintenance decisions solely on efficiency, but in a domain where it is impossible to reconstruct deleted cases, competence concerns should receive considerable weight.

8 Conclusion

An important current of CBR research studies how to develop strategies for achieving case-bases that are competent and compact, as a proxy for good system performance. This paper has presented an argument for integrating performance considerations more directly into case addition and deletion procedures, in order to allow finer-grained optimization of case-base contents. The paper shows that the relationship between competence, compactness and adaptation performance is more subtle than a simple tradeoff—in some circumstances, adaptation performance can be increased without sacrificing competence or compactness—motivating the search for ways to refine case addition and deletion procedures to improve performance results. It also presents empirical studies demonstrating relationships between competence criteria, adaptation performance, and case-base size, as well as an initial step towards developing a performance-guided metric for estimating the performance value of adding a case to a case-base.

Much remains to be done in refining this approach and providing a richer model. Such work includes refining the performance metric; performing more theoretical and empirical analyses of the tradeoffs and factors involved, considering both retrieval and adaptation costs; and combining competence and performance metrics to achieve metrics that balance both factors as desired. However, we believe that just as the direct connection of retrieval criteria to adaptation abilities led to important progress (Smyth & Keane 1998), the direct connection of case-base construction to performance criteria promises important advances for case-base maintenance research.

References

Chomsky, N. 1965. *Aspects of the Theory of Syntax.* Cambridge, MA.: MIT Press.

Doyle, M., and Cunningham, P. 1999. On balancing client-server load in intelligent web-based applications involving dialog. Technical Report TCD-CS-1999-25, Trinity College Dublin.

Francis, A., and Ram, A. 1993. Computational models of the utility problem and their application to a utility analysis of case-based reasoning. In *In Proceedings of the Workshop on Knowledge Compilation and Speed-Up Learning.*

Hart, P. E. 1968. The condensed nearest neighbor rule. *IEEE Transactions on Information Theory* 14:515–516.

Leake, D., and Wilson, D. 1998. Case-base maintenance: Dimensions and directions. In Cunningham, P.; Smyth, B.; and Keane, M., eds., *Proceedings of the Fourth European Workshop on Case-Based Reasoning*, 196–207. Berlin: Springer Verlag.

Leake, D., and Wilson, D. 1999. When experience is wrong: Examining CBR for changing tasks and environments. In *Proceedings of the Third International Conference on Case-Based Reasoning*, 218–232. Berlin: Springer Verlag.

Portinale, L.; Torasso, P.; and Tavano, P. 1998. Proceedings of the thirteenth european conference on artificial intelligence. In *Dynamic Case Memory Management*, 73–77. Chichester: Wiley.

Portinale, L.; Torasso, P.; and Tavano, P. 1999. Speed-up, quality, and competence in multi-modal reasoning. In *Proceedings of the Third International Conference on Case-Based Reasoning*, 303–317. Berlin: Springer Verlag.

Smyth, B., and Cunningham, P. 1996. The utility problem analysed: A case-based reasoning perspective. In *Proceedings of the Third European Workshop on Case-Based Reasoning*, 392–399. Berlin: Springer Verlag.

Smyth, B., and Keane, M. 1995. Remembering to forget: A competence-preserving case deletion policy for case-based reasoning systems. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, 377–382. Montreal: IJCAI.

Smyth, B., and Keane, M. 1998. Adaptation-guided retrieval: Questioning the similarity assumption in reasoning. *Artificial Intelligence* 102(2):249–293.

Smyth, B., and McKenna, E. 1999a. Building compact competent case-bases. In *Proceedings of the Third International Conference on Case-Based Reasoning.* Berlin: Springer Verlag.

Smyth, B., and McKenna, E. 1999b. Footprint-based retrieval. In *Proceedings of the Third International Conference on Case-Based Reasoning.* Berlin: Springer Verlag.

van Someren, M.; Surma, J.; and Torasso, P. 1997. A utility-based approach to learning in a mixed case-based and model-based reasoning architecture. In *Proceedings of the Second International Conference on Case-Based Reasoning*, 477–488. Berlin: Springer Verlag.

Zhu, J., and Yang, Q. 1999. Remembering to add: Competence-preserving case-addition policies for case base maintenance. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence*. Morgan Kaufmann.