Case Provenance: The Value of Remembering Case Sources

David Leake and Matthew Whitehead

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

Abstract. Case-based reasoning systems routinely record the results of prior problem-solving, but not the *provenance* of new cases: the way in which the new cases were derived. This paper proposes the value of tracking provenance information in CBR, especially when timely feedback may not be available. It illustrates the use of provenance information with studies of the application of provenance information to guide case-base maintenance. Experiments with two data sets illustrate the benefit of using provenance to propagate maintenance and to target maintenance effort.

1 Introduction

In case-based reasoning (CBR), memory of prior problems and solutions plays a central role: new solutions are generated by retrieving and adapting prior solutions, and are added to the case library for future use. However, standard CBR systems do not remember the *provenance* of the cases in their case libraries: how those cases came to be. This paper proposes that the storage of simple provenance information can play a valuable role in CBR for estimating solution confidence and guiding case base maintenance.

When a case is provided to a CBR system externally, provenance information records the external source. When a CBR system generates a case internally, a minimal provenance trace records the case(s) from which it was generated; a richer approach could also record information such as the adaptation strategies used. Such information provides many potential opportunities for refining system performance. For example, provenance information on externally-provided cases may be a useful source of clues to the case's applicability [1] or its reliability [2]. As an illustration, an ethnographic study on remote naval troubleshooting support for sailors showed that not all cases captured were treated equally: the reliability of the sailor who captured problem information was a crucial concern to experts who consulted the cases later [3].

When a CBR system generates cases internally by case adaptation, both the cases taken as starting point and the adaptations used may affect the quality of solutions; unreliable adaptations may cause quality loss, decreasing expected quality in cases generated by long sequences of unreliable adaptations. Thus considering case derivations may provide useful clues to solution quality. Simple provenance-based reasoning may enable increasing system robustness to other problems as well. For example, the effects of learning in a CBR system may depend on the order of case presentation; a CBR

R.O. Weber and M.M. Richter (Eds.): ICCBR 2007, LNAI 4626, pp. 194–208, 2007.

system may learn different things from a single set of cases, based on case presentation order. Tracking the history of case generation provides data which may be used to detect and discount presentation-order effects.

This paper presents an argument for the importance of case provenance. It begins by considering an implicit assumption of much CBR research, that feedback will be available. It shows that this assumption may not always hold in practice, and that provenance can be a useful tool to help alleviate some of the problems caused by absent or delayed feedback. The paper then considers a wider set of motivations for studying case provenance, including guiding maintenance, which may be needed even if the system receives timely feedback at case generation time. The discussion of motivations is followed with a series of five experiments. The experiments first focus on feedback issues, studying the effects of delayed feedback on solution quality and the use of provenance to propagate feedback information that becomes available to related cases. They then examine issues related to quality loss through repeated adaptations, examining solution quality trends and the use of the number of adaptations as a predictor for cases likely to require maintenance. The results illustrate how provenance information can guide the case-base maintenance process.

2 The Fallacy of Feedback

Given the potential uses of case provenance information, it is interesting to consider why provenance has not been a routine consideration within CBR systems. One possible explanation for considering only cases, rather than cases' origins, is that early CBR research commonly assumed that the cases in the case-base were correct, due to the CBR system receiving feedback on the success of its solutions as they are generated. Feedback was seen as essential for successful CBR, to assure that the system would not be led astray by reapplying failed solutions. However, the feedback assumption merits re-visiting for two reasons. First, in practice, feedback may be delayed or even unavailable, making it desirable to increase the robustness of CBR in the face of missing feedback. Second, even in domains for which feedback appears to be available, it may be incomplete. The goodness of solutions may depend on multiple dimensions, with feedback available only for some of them. In such situations, robustness to incomplete feedback is desirable as well.

Missing Feedback: In contexts such as CBR systems which provide advice to end users, feedback may be hard to obtain. User feedback rates are notoriously low; for example, the annual report of one help desk reports an average response rate under 8% [4]. Even when feedback eventually will be available, the reasoner may need to act before feedback is provided. In asynchronous troubleshooting, there may be a lag of hours or days before a help desk receives the response to its advice on a new problem, during which time similar problems may need to be solved. In product design, there may be a time lag of months or even years before product use and maintenance reveal problems, during which time new designs must still be generated.

Lack of feedback can cause problems for a CBR system. For example, without feedback, a CBR system's conclusions from a given set of problems may be radically different depending on problem presentation order. As a simple example, consider the task

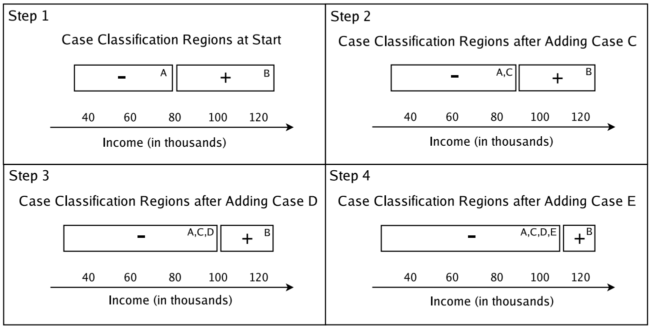


Fig. 1. Cases C, D, and E in sequence extend the negative classification region

of predicting loan eligibility based on the loan amount and the borrower's income. Assume that a CBR system predicts using 1-NN, with similarity determined by Euclidean distance, and that the system starts with seed cases A and B. Case A records a request for a \$20,000 loan, an income of \$40,000, and a negative decision; B records a request for a \$20,000 loan, an income of \$125,000, and a positive decision. Given the sequence of problems (C = (\$20,000, \$60,000), D = (\$20,000, \$80,000), E = (\$20,000, \$100,000)), cases C, D, and E will each extend the negative region, as illustrated in the 1-dimensional view of Figure 1. The same problems in the reverse order will successively extend the positive region, for the reverse effect. In either scenario, considering how the solutions were generated makes clear the need to treat the results with caution.

Incomplete Feedback: Even when feedback is available, it may be partial. In case-based planning, feedback may be available concerning a plan's success or failure, but not its comparative efficiency: The planner will know the number of plan steps but not whether alternative plans might have involved fewer steps. In an example from CHEF [5], the planner repairs an interaction problem by cooking two ingredients separately instead of together. If the resulting recipe is later modified, replacing the two ingredients with others which do not interact, they will be cooked separately, even if that is unnecessary. If a planner starts out with a set of high-quality expert plans, new plans generated with minor variations might be expected to have reasonable efficiency, but each successive adaptation may risk carrying forward aspects unneeded for the current situation and missing possible optimizations, regardless of whether feedback confirms successful accomplishment of goals. Here provenance information—how the new plan was generated from an expert plan—may be useful as a proxy for estimating aspects of quality not available from feedback to the system.

3 Motivations for Studying Case Provenance

While case provenance has not yet been studied as a CBR area in its own right, provenance considerations could contribute both to assessing case quality and to guiding case-base maintenance.

Provenance and Confidence: Recent research has observed the importance of methods for assessing confidence in the solutions of a CBR system. For example, Cheetham and Price [6] argue persuasively for the importance of internal methods for assessing confidence, and present an extensive set of confidence indicators, based on analyzing individual cases and their relationships (e.g., the sum of similarities of retrieved cases with the best solution). These provide rich criteria, provided that the cases in the case base are themselves assumed to be trustworthy. However, their trustworthiness depends on their own provenance.

In real-world case-based reasoning, cases may be collected from many distributed sources (e.g., [7]). Confidence in externally-provided cases may vary by source, making knowledge of sources important to balance tradeoffs between case similarity and source-based factors if one source is less reliable than another [1].

For internally-generated cases, confidence may depend both on the original cases and on their connections—the adaptation procedures generating one case from another. It is commonplace in rule-based systems to assign confidence values to rules, and to estimate the confidence of conclusions based on their derivations (e.g., [8]). For CBR systems, the quality of solutions may be estimated based on the quality of the original case and the chain of adaptation steps performed. In this paper, we explore the use of a very simple provenance-based metric for estimating quality, the length of the adaptation chain: the number of intermediate cases generated from an initial case before generating the current solution. This may enable estimating adaptation-based case quality decay for use in assessing confidence in a solution.

Provenance and Explanation: Beyond the direct use of provenance to assess confidence, provenance information may be useful to explanation of a CBR system's conclusions to the user. Understanding how a solution was derived from confirmed cases—perhaps through a chain of intermediate problem-solving—can provide users with a deeper understanding of how a solution was generated.

Provenance and maintenance: Case-base maintenance research has extensively examined case-base growth issues, focusing primarily on retention decisions for individual cases and factors such as consistency and coverage (for a sampling of this work, see [9]). This focus examines the contents of the cases in the case base at maintenance time, rather than their sources (one exception is the HOMER project [2], which distinguishes between cases captured directly from help desk operators and confirmed cases verified by a case author).

Tracking provenance information gives a new source of maintenance information, with many potential uses:

- *Responding to delayed feedback:* When feedback is delayed, an unconfirmed case may already have been used to solve other problems before its confirmation is received. If the original case is erroneous, the cases derived from it may require repair as well. Likewise, the error in the current case may suggest that the cases from which it was derived need repair as well. Provenance information enables identifying related cases for repair.
- *Focusing case-base maintenance effort:* In addition, internal case provenance information enables an analysis of the case-base's growth over time, and of the

influences cases have on each other over time. If labor-intensive methods are needed to maintain cases, cases which led to more adaptations are a natural candidate for confirmation; in instances of conflict, derivations may be useful as well, to check other regions of the case-base which are potentially affected.

– *Focusing similarity and adaptation knowledge maintenance effort:* Provenance information can suggest cases which may require attention, even in the absence of feedback. Conversely, when feedback is available and shows problems in a derived solution, provenance information about how the erroneous solution was generated can provide data to analyze for flaws in the system's similarity metric (if the case used as a starting point was a poor choice) or adaptation knowledge (if case quality decays quickly along paths involving particular adaptations).
– *Guiding maintenance based on trends as cases are applied:* There is a long history of CBR systems using feedback about problem solutions to repair the cases generated to solve them (e.g., [5]). The commonplace approach it that a new solution is generated, compared to feedback, and fixed if needed. Thus the repairs address the current solution, but assume that the previous case is correct and is a good precedent to use for similar cases.

  However, even correct cases may not be good precedents. For example, in a property value estimation domain, if the case for a particular house results in an erroneous estimate for a new problem, a new case, with the correct price for that problem, is stored; the initial case is retained unchanged. Nevertheless, if the original case repeatedly yields faulty predictions, the original case may require adjustment as well. If the previous house sold at an unusually high price, because the specific buyers were willing to pay a premium for personal reasons (e.g., proximity to their babysitter), using the case to predict the prices of other houses might often produce estimates which are too high. If a system retains information about both the cases to which a given case is adapted, and the success of those cases, analysis of this information could prompt repair of the case—e.g., in this example, an annotation to adjust how it is applied (e.g., "this case tends to suggest values 10% too high)—or an adjustment of the similarity metric in that region of the case base.

4 Experimental Design and Results

The previous section hypothesizes that maintenance guided by case provenance information can improve the overall performance of the CBR system. One way provenance information might be exploited is for automatic feedback propagation. Human experts can improve the quality of CBR systems by giving accurate reference solutions to cases already in the case base, but it may be infeasible for a human expert to correct a large number of cases. Therefore, we would like to maximize the benefit of each instance of feedback a human expert is able to give, by applying that feedback to improve the quality of related cases. Our experiments simulate a scenario in which human expert feedback completely corrects a solution for a single case in the case base and then that solution is used to repair the solutions for cases that were derived from the corrected case. This approach was implemented using IUCBRF [10], a freely-available Java case-based reasoning framework developed at Indiana University, extended with the needed maintenance functionality. The experiments explored the following questions:

- How do feedback delays affect overall solution quality?
- Is the length of the adaptation chain generating a case predictive of its solution quality?
- Is provenance-based maintenance propagation a beneficial strategy, and what are its computation costs?
- Is provenance information useful for selecting cases for which to solicit external feedback (e.g., from a human expert)?

Case Base Datasets and Setup. Our tests used two separate case bases, the Boston Housing Database and Abalone Database from the UCI [11] machine learning repository. The Boston Housing Dataset contains 506 cases, capturing attributes of house types in the Boston metro area. The dataset has one class attribute, the median price of houses of the given house type. In the experiments, the CBR system's goal was to predict median housing prices. Seed case bases for these experiments included 100 house types chosen at random, with the test sets composed of the remaining house types. The Abalone Dataset includes 4177 cases with one class attribute, the age of the abalone, which is continuously valued. This dataset was used to populate case bases with 100 cases along with their reference solutions, with the other data points used for testing. A new case base was generated for each trial run.

The problems presented to the CBR systems were solved by adapting prior cases using simple heuristics. For the Boston Housing Database, new solutions were formed by taking the case with the most similar problem features and offsetting its median house price by the relative difference in the sizes of the houses. A similar technique was used for the Abalone Age Dataset using the age of the nearest neighbor case and the relative lengths of the abalones.

Provenance Information Used. As our testbed system generates new cases by adaptation, it records the cases from which new cases are derived. This information is used to define the following relationships, considered by provenance-based maintenance processes: Case C is a *child* of *parent* case P if C was generated by adapting P; case D is a *descendant* of *ancestor* case A if case D was generated through some chain of adaptations from A (either a single adaptation or adaptations through a chain of intermediate cases). Any *descendant* or *ancestor* of C is considered *related* to C.

4.1 Test 1: Solution Quality with Delayed Feedback

The first experiment measured the solution quality decay of a CBR system given feedback delayed by various time intervals. At each time step a new problem was presented to the system to solve. The new adapted case, C, was then added to the case base, with a case removed at random if the case base size limit was exceeded. After n steps, feedback in the form of a reference solution was given for C and adapted solutions were propagated to all related cases.

We used the values $1, 5, 10, 50$ as the number of steps of delay before giving feedback to the system. The mean absolute error (MAE) average of all the cases in case base was graphed at each time step.

Test 1: Results. Figure 2 illustrates the dramatic benefit of feedback in the sample domains. Each plot line starts with increasing error up until the problem number when expert feedback is first received. At this point the error for each stops increasing and gradually decreases over the remainder of the problems. We note that the initial slope, while error increases, depends on the effectiveness of the adaptation method; better adaptation methods will yield gentler slopes, meaning less error added per adaptation.

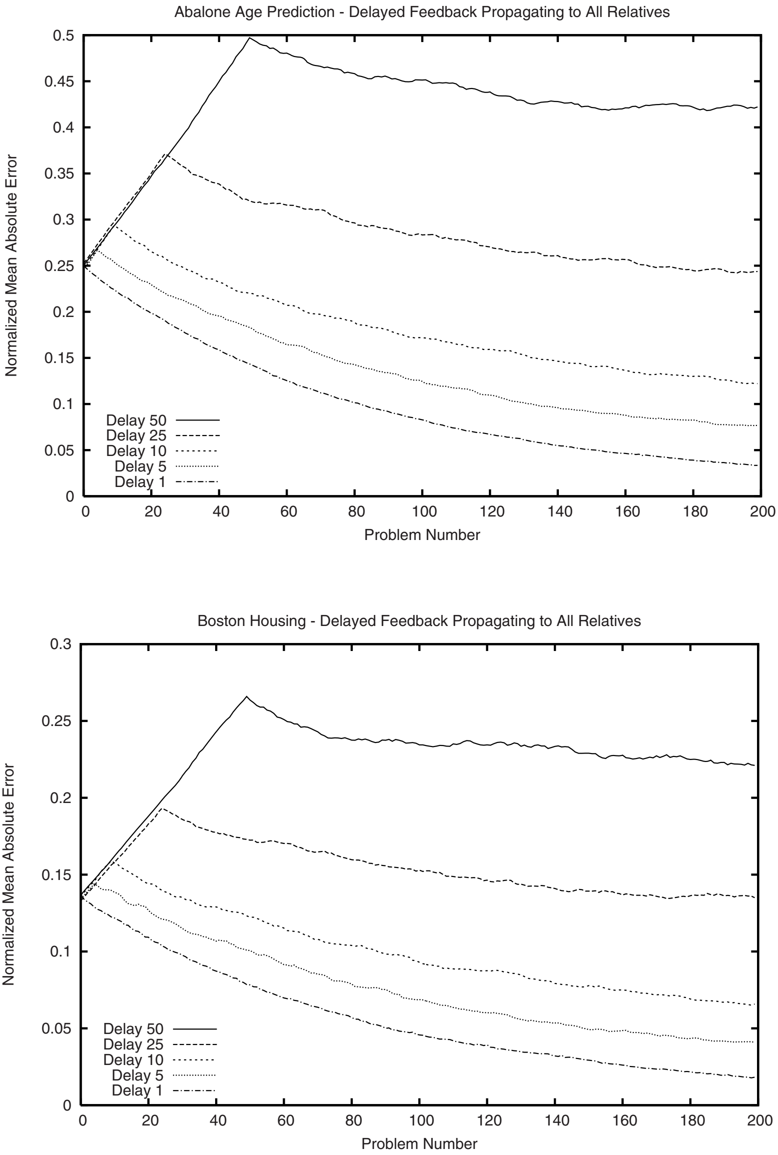


Fig. 2. MAE with varying feedback delays for the Abalone Age and Boston Housing datasets

4.2 Test 2: MAE Per Adaptation

This test measured the amount of error introduced into the case base for each adapted solution that was added. For this test we initialized case bases with 100 reference cases and then presented them 1000 randomly chosen new problems. The CBR system solved each problem, added the newly created case for the problem, and then performed maintenance by randomly deleting a case. No feedback was used for this test.

After each problem was solved, the MAE was computed for the entire case base along with the average number of adaptation generations for all the cases. These two values were stored after each problem presentation and later graphed to show the relationship between error and the number of adaptation generations.

Test 2: Results. Results were similar for both case bases, so only the results from the Boston Housing dataset are shown. Figure 3 shows the relationship between a case base's average number of adaptation generations per case and the normalized MAE for the entire case base. As expected, the greater the number of adaptation generations, the higher the overall error for the case base. As with Test 1, the slope of the linear fit line reflects the effectiveness of the system's adaptation method.

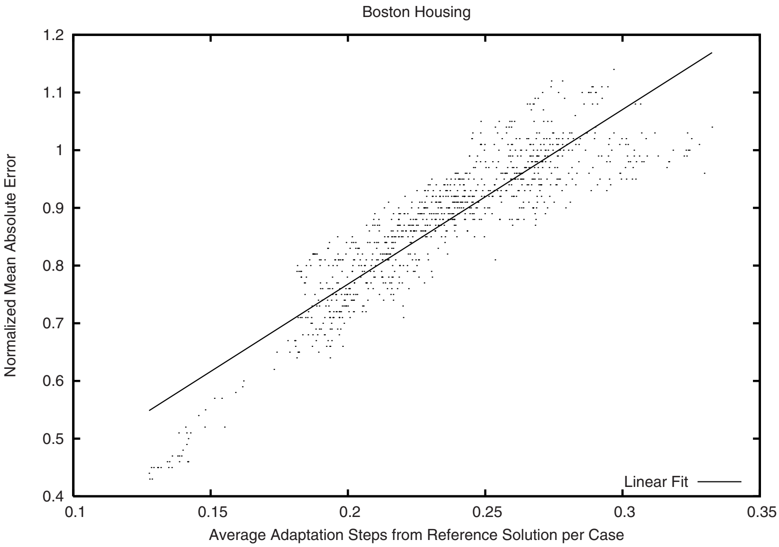


Fig. 3. The average number of adaptation generations per case in a case base is directly related to the case base's overall quality in the Boston Housing Database

4.3 Test 3: Using Feedback Propagation to Improve Case Base Quality

The third test examined different feedback propagation strategies using case provenance information, using MAE of all the cases in the case base to measure solution quality.

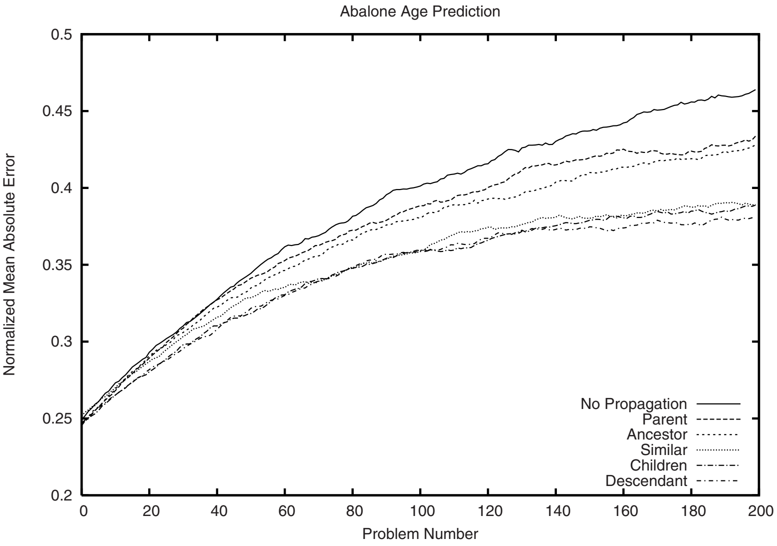
For each trial, the case base was initialized with 100 randomly-selected cases with known correct reference solutions, and a series of 200 test problems was presented to the system to solve. For each test problem, the system retrieved the most similar stored case and adapted its solution to the current problem. The adapted solution and current problem then formed a new case that was inserted back into the case base. Case base size was limited to 100 cases throughout the tests, with a randomly-chosen case removed from the case base for each addition, to keep the case base size constant. The entire test was repeated 100 times and the resulting MAE values were averaged over all the runs.

After each test problem, feedback was given for a single random case R in the case base. This was meant to simulate a human expert giving the system feedback. Once the feedback was given, one of the following solution propagation strategies was applied.

- *No Propagation:* Only the single case, R, that was given the reference solution was changed. This is the baseline method that has been common in past CBR systems.
- *Propagation to Similar Cases:* The entire case base is searched for cases that are similar to R within a given similarity threshold T. Sufficiently similar cases are then given new solutions adapted directly from R's reference solution. In our tests T was 0.2 for the Boston Housing dataset and 0.1 for the Abalone Age dataset.
- *Propagation to Children:* Any child cases of R are given new solutions adapted directly from R's reference solution.
- *Propagation to Parent:* Any parent case of R is given a new solution adapted directly from R's reference solution.
- *Propagation to Descendants:* Any descendant cases of R are given new solutions adapted from their immediate parent cases. This corresponds to recursively adapting solutions through generations of descendants from R's reference solution.
- *Propagation to Ancestors:* Any ancestor cases of R are given new solutions adapted from their immediate children cases. This corresponds to recursively adapting solutions up through generations from R's reference solution.

Test 3: Results. Figure 4 shows the results of different types of feedback propagation for the two data sets. No propagation results in the highest error across the problems (approximately 0.46 normalized MAE), i.e., every form of feedback propagation helped decrease overall error to some degree. *Propagation to children* and *propagation to descendants* outperformed *propagation to parents* and *propagation to ancestors*, which can be attributed to the greater number of cases reached by propagation to children and descendants: multiple cases may be adapted from a single parent case but each case has only a single parent. Overall, automatically propagating feedback to relative cases appears promising. The best performing feedback propagation methods reduced the error between 12% and 17% for the two test case bases.

On the Abalone dataset, performance of *propagation to similar cases* was nearly equivalent to *propagation to children*, but was only roughly comparable to *propagation to parent* and *propagation to ancestor* for the Boston Housing dataset. This difference is



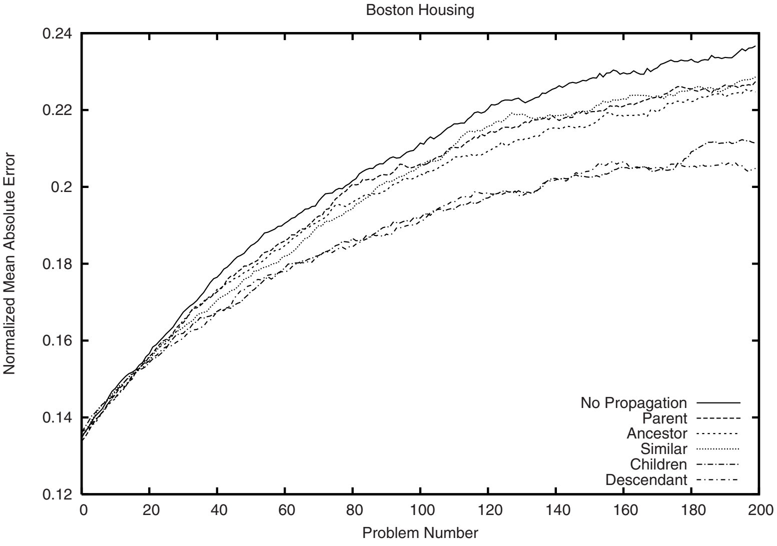


Fig. 4. MAE for varying propagation methods for the Abalone Age and Boston Housing datasets

a subject for further study. It may be attributable to inherent differences in the datasets or to the similarity thresholds used to determine which cases in the case base were considered similar enough to be given new solutions. However, additional tests adjusting the similarity thresholds were not conclusive.

4.4 Test 4: Computational Efficiency of Feedback Propagation

Propagating feedback by adapting case relatives increases processing cost, raising the question of tradeoffs between propagation time and improvement in solution quality for candidate methods. To examine this tradeoff, we compared feedback propagation time for each method. Propagation time was defined as CPU time (on a 1.2 GHz Pentium 4 with 768 MB of RAM) from presentation of feedback until completion of all case base updating. This test used case bases with 100 cases for each dataset and ran 1000 randomly chosen problems selected from the case bases with replacement. We then graphed the propagation times for each different technique.

Test 4: Results. Figure 5 shows the feedback propagation times of the various propagation methods for both datasets. All the propagation methods took less than 2 seconds per 1000 test problems. The method of propagating feedback to any similar cases (whether directly adapted from the corrected case or not) took substantially more time than the other methods, due to identifying similar cases by linear search through the case base. Indexing strategies could significantly decrease this cost, but provenance-based methods might still be preferable to similarity-based methods for very large case bases.

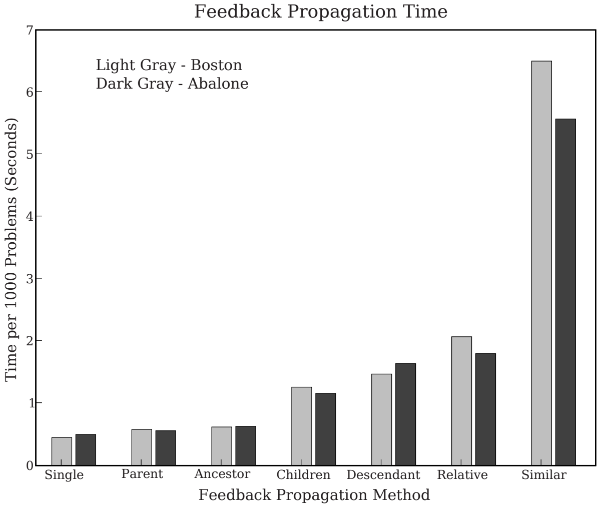


Fig. 5. Computational overhead of feedback propagation methods

4.5 Test 5: Targeted Feedback

Section 3 hypothesized that provenance information may be useful for predicting case quality in systems for which quality is expected to decay with repeated adaptations. Similarly, provenance information may be useful for directing limited maintenance resources, e.g., by directing a human expert towards those cases that when updated will maximize the benefit to the CBR system. This is especially important when the feedback process is time-consuming and costly. If the system can identify which cases are likely to have ineffective solutions, then the expert can focus on correcting those cases first, with the aim of maximizing the benefit of human effort. Likewise, if verifying a case requires additional costs (e.g., for running tests, etc.), the ability to target the right cases may be valuable.

For this test, recorded provenance information included a count of the length of the provenance path, i.e., the number of adaptations that a particular case is from a known accurate solution. We expected that given the imperfect case adaptation strategy used, cases closer in lineage to accurate solutions would be more accurate themselves, due to compounding of errors as repeated adaptations are performed.

The test compared resulting quality with four different maintenance techniques. The first, the baseline, used no feedback at all. The second, in each trial, used feedback to correct a random case in the case base. The third requested feedback on the case in the case base which was the highest number of generations from a reference solution, and corrected that case. The fourth corrected the case with the maximum error. We used 100 randomly chosen problems for each technique on case bases of size 100. We ran 25 trials and averaged the MAE for each problem.

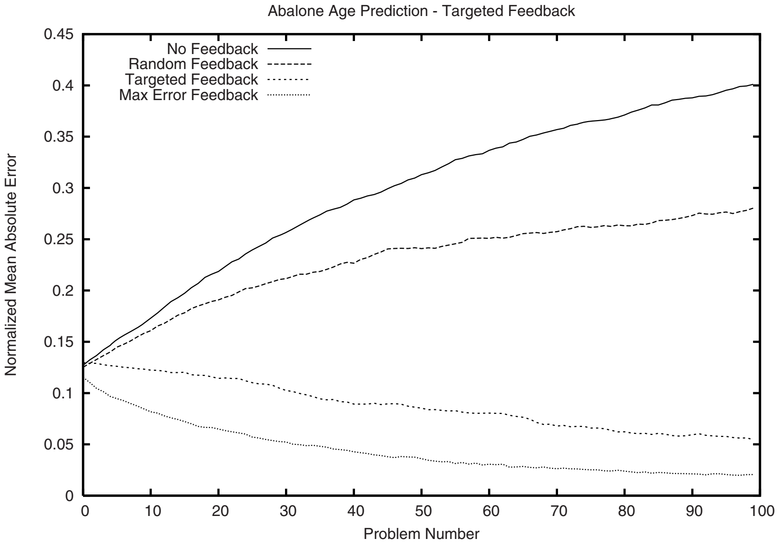
Test 5: Results. Figure 6 shows that targeting feedback towards cases with the longest adaptation chain substantially improves overall solution quality, compared to randomly picking cases for feedback. This can be explained by the feedback improving the cases expected to account for the greatest error.

For the two datasets, targeting based on adaptation history reduced the error obtained using random feedback by 75% on the Abalone dataset, and 82% on the Boston dataset. This suggests the value of targeted feedback, and that the number of adaptations performed provides a useful proxy for identifying cases with the most error, when the actual amount of error is not known. For comparison, the bottom line on each graph shows the effect which would be achieved with the optimal strategy of always correcting the case with greatest error.

Other methods for targeting feedback are an interesting topic for future research. For example, targeting feedback to the case with the most descendants and then using the corresponding propagation method from Test 3 might provide additional benefits.

4.6 General Observations

Overall, the tests are encouraging for the use of case adaptation history information to guide maintenance for systems with weak adaptation. If a CBR system already has a very accurate adaptation method, then there is little error introduced per adaptation and the feedback propagation and targeting methods do not have as dramatic an impact. If a CBR system has a poor method of adaptation that on average adds substantial error per adaptation, then the methods tested above are even more beneficial.



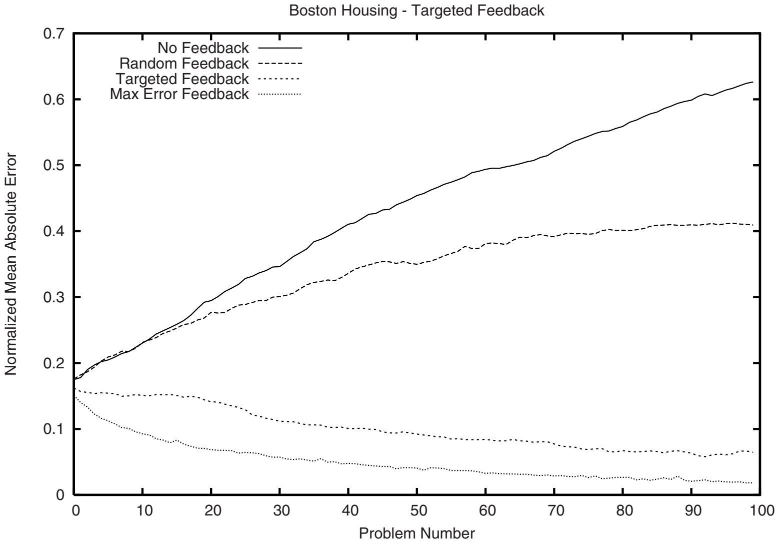


Fig. 6. Effect of selection of cases to correct on MAE for the Abalone Age and Boston Housing datasets

5 Related Work

The general notion of provenance is now attracting much attention in the e-Science community [12], for tasks such as enabling replication of results and estimating quality of scientific data. It is also attracting interest in Semantic Web research, for example, to support explanation (e.g., [13]). Tracking the derivations of beliefs and using those derivations to guide belief updating has a long history in AI, dating back to work on truth maintenance systems [14].

In the CBR literature, Goel and Murdock [15] proposed meta-cases to capture the reasoning underlying the CBR process, to support explanation of the reasoning underlying the processing of an individual case; such a reasoning trace is stored by the ROBBIE system as well [16]. However, in both these systems, the focus is on applying the trace to understand current reasoning, rather than understanding the extended derivation history of a case through the chain of cases from which it was derived.

6 Conclusion

This paper has argued for the value of studying of case provenance, and has illustrated the potential value of provenance-based strategies for estimating case confidence and guiding maintenance. The provenance-based approach is innovative in that—unlike maintenance work which only detects and fills gaps, or responds to problems revealed by feedback or inconsistencies—provenance-based methods can make *a priori* suggestions of candidates for case replacements or confirmations.

The paper explores simple strategies with much room for refinement. Interesting questions include how to use richer provenance information, such as information on the specific adaptations performed, and how to exploit such information for finer-grained prediction of case quality and for case base maintenance propagation strategies.

Provenance considerations may also prove useful for explanation, to enable grounding explanations of new solutions in authoritative cases connected to the current situation by short adaptation chains. The CBR community has long noted the value of supporting a conclusion by the known prior case from which it is derived (e.g., [17]). However, when solutions are based on cases generated by the system, simply showing the prior system case may not be as compelling. An interesting question is whether user trust may be increased by showing the full derivation of a solution, back to an externally-provided or externally-confirmed case.

Acknowledgment

We thank Steven Bogaerts and the anonymous reviewers for helpful comments.

References

1. Leake, D., Sooriamurthi, R.: Case dispatching versus case-base merging: When MCBR matters. International Journal of Artificial Intelligence Tools 13(1), 237–254 (2004)
2. Göker, M., Roth-Berghofer, T.: Development and utilization of a case-based help-desk support system in a corporate environment. In: Althoff, K.-D., Bergmann, R., Branting, L.K. (eds.) Case-Based Reasoning Research and Development. LNCS (LNAI), vol. 1650, pp. 132–146. Springer, Heidelberg (1999)

3. Evans, M.: Knowledge and Work in Context: A Case of Distributed Troubleshooting Across Ship and Shore. PhD thesis, Indiana University (2004)
4. Partington, N.: CIS helpdesk feedback statistics (2006) last modified (October 2, 2006), Accessed at http://www.ljmu.ac.uk/cis/aboutCIS/79910.htm
5. Hammond, K.: Case-Based Planning: Viewing Planning as a Memory Task. Academic Press, San Diego (1989)
6. Cheetham, W., Price, J.: Measures of solution accuracy in case-based reasoning systems. In: Funk, P., González Calero, P.A. (eds.) ECCBR 2004. LNCS (LNAI), vol. 3155, pp. 106–118. Springer, Heidelberg (2004)
7. Borron, J., Morales, D., Klahr, P.: Developing and deploying knowledge on a global scale. In: Proceedings of the Thirteenth National Conference on Artifical Intelligence, vol. 2, pp. 1443–1454. AAAI Press, Menlo Park, CA (1996)
8. Shortliffe, E.: Computer-based medical consultations: MYCIN. American Elsevier, New York (1976)
9. Leake, D., Smyth, B., Wilson, D., Yang, Q. (eds.): Maintaining Case-Based Reasoning Systems. Blackwell (2001) Special issue of Computational Intelligence, 17(2) (2001)
10. Bogaerts, S., Leake, D.: IUCBRF: A framework for rapid and modular CBR system development. Technical Report TR 617, Computer Science Department, Indiana University, Bloomington, IN (2005)
11. Blake, C., Merz, C.: UCI repository of machine learning databases (2000), http://www.ics.uci.edu/~mlearn/MLRepository.html
12. Simmhan, Y., Plale, B., Gannon, D.: A survey of data provenance in e-science. SIGMOD Record 34(3), 31–36 (2005)
13. Murdock, J., McGuiness, D., da Silva, P.P., Welty, C., Ferrucci, D.: Explaining conclusions from diverse knowledge sources. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 861–872. Springer, Heidelberg (2006)
14. Doyle, J.: A truth maintenance system. Artificial Intelligence 12, 231–272 (1979)
15. Goel, A., Murdock, J.: Meta-cases: Explaining case-based reasoning. In: Smith, I., Faltings, B.V. (eds.) Advances in Case-Based Reasoning. LNCS, vol. 1168, pp. 150–163. Springer, Heidelberg (1996)
16. Fox, S., Leake, D.: Modeling case-based planning for repairing reasoning failures. In: Proceedings of the 1995 AAAI Spring Symposium on Representing Mental States and Mechanisms, pp. 31–38. AAAI Press, Menlo Park, CA (1995)
17. Cunningham, P., Doyle, D., Loughrey, J.: An evaluation of the usefulness of case-based explanation. In: Ashley, K.D., Bridge, D.G. (eds.) ICCBR 2003. LNCS, vol. 2689, pp. 122–130. Springer, Heidelberg (2003)