Maintenance of Case Contents and Adaptation Rules

Brian Schack
Advisor: Dr. David Leake
Indiana University, Bloomington IN 47408, USA
schackb@indiana.edu

Abstract. This research summary outlines and addresses three problems in case-based maintenance on case contents and adaptation rules: 1) how to perform maintenance on divisible cases with non-uniform sizes, 2) how can adaptation knowledge improve the performance of maintenance on structured cases, 3) how to determine and use coverage for adaptation rules. Evaluation showed that, for suitable cases bases, maintenance strategies that subdivide cases and employ adaptation knowledge can outperform per-case strategies. A planned experiment will expand or contract the coverage claimed by adaptation rules and measure the effects on problem-solving performance. The conclusion summarizes research progress to date and areas for further research.

Keywords: case-based reasoning, case-base maintenance, flexible feature deletion, adaptation-guided maintenance, adaptation knowledge

1 Introduction

Case-based reasoning is a method of machine learning for solving problems involving four phases: retrieval, reuse, revision, and retention [1]. Its overall performance depends in no small part on the case base. Even starting with a high-quality case base, the passage of time motivates the need for case-base maintenance. Over time, the system will solve problems and store their solutions in its case base. As these solutions accumulate, they take up storage space, and they take time to search through. The passage of time can also make stored cases in need of revision or obsolete entirely. Over time, even the case representation can change as the system learns more about its domain or its environment changes.

2 Flexible Feature Deletion

My research on flexible feature deletion started by questioning the assumptions for case-based maintenance [3]. First, nearly universally, the evaluations of case-based maintenance strategies assume a uniform size for cases. Although correct for many representations, this assumption does not hold for variable-length feature vectors or more complex structured representations. For example, a case base of films could have different sizes depending on their running times or their numbers of actors and actresses.

This suggests that maintenance strategies employing deletion should consider not only the coverage benefit provided by a case, but also the storage cost of retaining it.

A second assumption states that cases do not permit subdivision of their contents. Normally, a maintenance strategy will either choose to delete or retain an entire case. But if contents permit sub-deletion, abstraction [4], or alteration, then the size of a case base can change independently from the number of cases in it. For example, a case base of medical imagery [5] could delete irrelevant regions or represent them at a lower level of detail.

Dismissing both of these assumptions allows for a classification of different kinds of maintenance strategies depending on how they subdivide the case base: case-bundled, feature-bundled, and unbundled. First, case-bundled strategies delete an entire case including its component features. Second, feature-bundled strategies delete a single feature across all of the cases in the case base. And third, unbundled strategies delete individual a case-feature pair independently of the remainder of the case to which it belongs or the occurrences of the same feature in other cases.

Along those lines, I implemented 11 simple maintenance strategies. The strategies are named according to what they target for deletion first. For example, the Rarest Features strategy deletes features in order from the rarest to the most common. Three hybrid strategies combine pairs of strategies with a 50/50 weighting. The following table compares each of the maintenance strategies:

Strategy	**Type of Bundling**	**Hybrid or Non-Hybrid**
Random Case-Features	Unbundled	Non-Hybrid
Random Cases	Case-Bundled	Non-Hybrid
Largest Cases	Case-Bundled	Non-Hybrid
Least Coverage	Case-Bundled	Non-Hybrid
Most Reachability	Case-Bundled	Non-Hybrid
Random Features	Feature-Bundled	Non-Hybrid
Rarest Features	Feature-Bundled	Non-Hybrid
Most Common Features	Feature-Bundled	Non-Hybrid
Largest Cases / Least Coverage	Case-Bundled	Hybrid
Rarest Features / Least Coverage	Unbundled	Hybrid
Rarest Features / Large Cases	Unbundled	Hybrid

I evaluated the strategies on case bases across three different domains: IMDb films, Congressional bills, and travel agency packages. As always, there is no such thing as a free lunch because of the trade-off between accuracy and size [6]. The question was then which strategy could retain the most competence in spite of deletions. The evaluation showed that the best strategy varied depending on the case base, but for some cases bases, unbundled and feature-bundled strategies could outperform case-bundled strategies.

3 Adaptation-Guided Maintenance

The results for the simple strategies inspired me to ask whether more powerful strategies could achieve a higher level of performance. I looked for sources of knowledge to bring to bear, and adaptation knowledge seemed the most promising [7]. The adaptation phase revises internal case contents in order to make the retrieved solution more suitable to the given problem. And feature-level maintenance also revises case contents, but in this situation, in order to reduce case base size. So, in a sense, both the adaptation and maintenance phases perform adaptations (perhaps even from the same set of possibilities) just with differing goals.

Therefore, I investigated whether maintenance could tie its deletion decisions directly to adaptation knowledge in order to improve on flexible feature deletion. Furthermore, the maintenance strategy could prioritize a deletion of a component within a case according to its recoverability through further adaptations or chains of adaptations. From a different perspective, this approach deletes knowledge overlapping between the case and solution transformation containers [8].

I evaluated this idea in a path planning domain with the goal of finding the shortest path between vertices on a weighted graph representing a route between waypoints on a network of roads. The following table shows the maintenance strategies employed:

Maintenance Strategy	**Lossiness**	**Description**
Shared Component	Lossless	Extract components shared by the solutions of multiple cases into separate cases. Mark gaps for completion during recovery.
Reachability-Based Largest Case	Lossy	Delete cases in order from largest to smallest number of case-features, deleting only recoverable cases.
Largest Case	Lossy	Delete cases in order from largest to smallest number of case-features regardless of recoverability.
Recoverability-Based Random Vertex	Lossy	Delete randomly-chosen case-features from the solutions to cases, deleting only recoverable features.
Random Vertex	Lossy	Delete randomly-chosen case-features from the solutions to cases regardless of recoverability.

Evaluation showed that the Shared Component maintenance strategy retained the most competence as expected because of its classification as lossless. But its compression ability maxed out at about 70% of the size of the uncompressed case base when it could not find any more shared components. Among the lossy strategies, recoverability-based largest case performed next best which showed that the recoverability-based approach can improve competence retention by using adaptation knowledge.

As mentioned earlier, normally competence always decreases with increased compression. However, the results of this experiment surprised me because occasionally adaptation-guided maintenance slightly improved the competence of the system by subdividing cases to make their components accessible to adaptations of limited power – – a phenomenon that I referred to as *creative destruction*.

4 Adaptation Knowledge Coverage

For my current research topic, I considered building on the creative destruction idea. I think I can show theoretically how a formal system could use the same rewriting rules for both adaptation and maintenance, and with suitable rules, creative destruction could have a significant effect. Unfortunately, I haven't found an appropriate domain in which to apply and evaluate this practically.

I settled on the topic of adaptation knowledge coverage. Much research on maintenance has highlighted the importance of the coverage of cases [9], but on the other hand, our field knows comparatively less about how to determine and use coverage for adaptation knowledge. I'm working with a real estate case base consisting of houses for sale with features for their prices, number of bedrooms, square feet, etc. I did not find off-the-shelf adaptation rules for this domain, so I developed a system for learning the rules from pairs of cases (as others have done before me). Together the case base and the learned adaptation rules form an oracle.

Next, I intend to make a copy of the adaptation rules by removing contextual restrictions so that they conflict with one another. I'll eagerly apply rules to the cases and ask the oracle to judge the quality of the derived cases. This will generate triples of case, rule, and quality. From this, I can judge the reliability of the rules and delete the least reliable rules.

Going further, I can look for common features between cases where the same rule applies with a high quality and then restrict the rule to those features. Or alternatively, common features between cases where the same rule applies with a low quality, and then restrict the rule to the negation of those features. The claimed coverage of an adaptation rule could exceed its actual coverage or vice versa. To evaluate this, I'll compare the performance of the oracle, maintained adaptation rules, and unmaintained rules.

5 Further Research

Maintenance necessarily involves a three-fold trade-off between problem-solving competence, problem-solving time, and storage space. Normally, there is no free lunch because reductions in size will also reduce competence. But a lossless maintenance strategy can reduce size to a limited extent without competence reduction, and creative destruction can occasionally even improve competence. Normally, reductions in size mean less cases to search through and therefore reduced retrieval time. But the increased adaptation time to recover a usable solution could cancel out the reduced retrieval time.

Additionally, similarity metrics can involve (perhaps recursively) comparing case components either one-to-one or even many-to-many. Deletion of case components could decrease (or in some comparisons, increase) the time taken by the similarity metric. For example, [10] uses feature reduction to balance the trade-off between the benefit of keeping a feature and the cost of similarity comparisons involving it. In future research, I'd love to explore the different directions of the competence-time-space trade-off on flexible feature deletion and its dependence on the properties of different case bases.

In my research, I used different domains for the different experiments to show broad applicability. But ultimately, I'd like to tie together all of the strategies into a single experiment in order to measure the relative benefit of each separately and together.

6 Conclusion

In conclusion, my research focuses on the maintenance phase of the case-based reasoning cycle. I dismissed the assumptions of uniform size and indivisibility of cases yielding flexible feature deletion strategies. Then, I incorporated adaptation knowledge into these strategies and applied them to structured cases. Next, I intend to continue studying adaptation knowledge especially how to determine the limits of its coverage and how knowledge of coverage for adaptation rules can improve maintenance strategies.

References

1. Aamodt, A., & Plaza, E. (1994). Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI communications*, *7*(1), 39-59.
2. Francis, A. G., & Ram, A. (1993). The utility problem in case-based reasoning. In *Case-Based Reasoning: Papers from the 1993 Workshop* (pp. 160-161).
3. Leake, D., & Schack, B. (2015, September). Flexible feature deletion: compacting case bases by selectively compressing case contents. In *International Conference on Case-Based Reasoning* (pp. 212-227). Springer International Publishing.
4. Bergmann, R., & Wilke, W. (1996). On the role of abstraction in case-based reasoning. *Advances in case-based reasoning*, 28-43.
5. Wilson, D. C., & O'Sullivan, D. (2008). Medical imagery in case-based reasoning. In *Case-Based Reasoning on Images and Signals* (pp. 389-418). Springer Berlin Heidelberg.
6. Lupiani, E., Craw, S., Massie, S., Juarez, J. M., & Palma, J. T. (2013, July). A multi-objective evolutionary algorithm fitness function for case-base maintenance. In *International Conference on Case-Based Reasoning* (pp. 218-232). Springer Berlin Heidelberg.
7. Leake, D., & Schack, B. (2016, October). Adaptation-Guided Feature Deletion: Testing Recoverability to Guide Case Compression. In *International Conference on Case-Based Reasoning* (pp. 234-248). Springer International Publishing.
8. Richter, M. M. (2003). Knowledge containers. *Readings in Case-Based Reasoning. Morgan Kaufmann Publishers*.
9. Smyth, B., & Keane, M. T. (1995, August). Remembering to forget. In *Proceedings of the 14th international joint conference on Artificial intelligence* (pp. 377-382).
10. Floyd, M., Davoust, A., & Esfandiari, B. (2008). Considerations for real-time spatially-aware case-based reasoning: A case study in robotic soccer imitation. *Advances in Case-Based Reasoning*, 195-209.