Exploration vs. Exploitation in Case-Base Maintenance: Leveraging Competence-Based Deletion with Ghost Cases

David Leake(✉) and Brian Schack

School of Informatics, Computing, and Engineering, Indiana University,
Bloomington, IN 47408, USA
{leake,schackb}@indiana.edu

Abstract. Case-base maintenance research has extensively studied strategies for competence-retaining case base compression. Such approaches generally rely on the *representativeness assumption* that current case base contents can be used as a proxy for future problems when determining cases to retain. For mature case bases in stable domains, this assumption works well. However, representativeness may not hold for sparse case bases during initial case base growth, for dynamically changing domains, or when a case base built for one task is applied to cross-domain problem-solving in another. This paper presents a new method for competence-preserving deletion, Expansion-Contraction Compression (ECC), aimed at improving competence preservation when the representativeness assumption is only partially satisfied. ECC precedes compression with adaptation-based exploration of previously unseen parts of the problem space to create "ghost cases" and exploits them to broaden the range of cases available for competence-based deletion. Experimental results support that this method increases competence and quality retention for less representative case bases. They also reveal the unexpected result that ECC can improve retention of competence and quality even for representative case bases.

Keywords: Case base maintenance · Competence
Knowledge containers · Representativeness assumption

1 Introduction

Much case-based reasoning research focuses on how to develop compact, competent case bases (*e.g.*, [1,2,7,19,27,34,38]. The desire for compact, competent case bases arose from retrieval efficiency concerns (*e.g.,* [9,32,35]). Such efficiency concerns remain an issue for CBR applied to big data, and case base compression remains important for other reasons as well. Compact case bases are easier for humans to maintain, and compact case bases may facilitate knowledge sharing

M. T. Cox et al. (Eds.): ICCBR 2018, LNAI 11156, pp. 202–218, 2018.
https://doi.org/10.1007/978-3-030-01081-2_14

and reasoning about the competence of other agents in distributed case-based reasoning [30].

Many methods have been developed for controlling case base growth. Because case deletion may result in unrecoverable knowledge loss, a central focus has been selective deletion aimed at maximum competence preservation, starting with seminal work by Smyth and Keane [33] and continuing with many other methods (*e.g.*, [19,20,27,34]). Retention methods generally estimate the competence contribution of each case, to prioritize retention decisions according to maximum competence contributions. Estimating the future competence contributions of a case is difficult because it depends on predicting the problems a CBR system will encounter. Smyth and McKenna proposed addressing this with the *representativeness assumption* [34] that the prior problems are representative of the problems to be encountered in the future. Under this assumption, the future competence contribution of a case can be estimated as its competence contribution in the existing case base. Although the assumption may not always hold, Smyth and McKenna advanced a compelling argument for its appropriateness for CBR systems: Because CBR is based on the assumption that future problems will resemble previous problems (problem-distribution regularity [23]), domains for which the representativeness hypothesis fails would be ill-suited for CBR.

Case-base compression relying on the representativeness assumption has been shown effective in many domains. However, in domains for which only a small part of the problem space has yet been encountered, or in which concept drift shifts the problem distribution [5], representativeness may not hold, and in turn, competence may suffer [26]. Likewise, if a case base generated for one task is applied to a new task for cross-domain problem-solving [21], there is no guarantee that the problems of the first space will be representative of problems in the second.

A well-known strength of CBR is that it can draw on multiple knowledge containers whose contributions overlap, in the sense that strengths in one can compensate for weaknesses in another [31]. This paper investigates how a CBR system can draw on adaptation knowledge to handle experience gaps when building a case base. By adapting cases already in the case base, a CBR system can pre-populate sparsely populated regions of the case base—transferring some of the knowledge of its adaptation component into the case component to expand the set of cases. This in turn enables the system to generate the compressed case base from a set of candidates larger than its retained experience. This can be seen as shifting from maintenance that only exploits existing experiences, to maintenance that explores the space of future problems. Combining case base exploitation with problem space exploration is a novel step for case-base maintenance.

To combine exploitation and exploration for case-base compression, this paper proposes the new method *expansion-contraction compression* (ECC). ECC adapts existing cases to generate additional candidate cases, which we call "ghost cases," providing a more diverse set of cases for compression to consider. In ECC, the union of the case base and ghost cases is then provided to the condensed nearest neighbor algorithm (CNN) [11] in order of competence contribution. This provides competence-based deletion with a wider set of cases from which to select that can include cases from unseen but solvable parts of the problem space. If case adaptation is considered sufficiently reliable, the result of ECC can be used as-is. Otherwise, the selected ghost cases can become targets for verification, *e.g.*, by asking a human expert in an active learning process, or provenance information [22] about the origin of ghost cases could be used to predict confidence when they are used.

This paper presents an evaluation of ECC for four standard data sets manipulated to enable controlled comparisons with CNN for case bases with varying representativeness. We hypothesized that ECC would provide better competence retention than CNN as representativeness decreased and observed this in the results. We also hypothesized that ECC would not provide benefit for standard case bases and would even impose a competence penalty, due to ghost cases increasing the coverage density for non-representative problems at the expense of coverage density for representative problems. Surprisingly, however, ECC often improved competence retention even for standard case bases. We attribute this to the addition of ghost cases providing CNN with more extensive choices, enabling it to select a more effective mix of cases.

2 Background

2.1 Compressing the Case Base

A primary early motivation for case-base compression was the swamping utility problem for CBR (*e.g.,* [9,32,35]). As the case base grows, case retrieval costs generally increase, while case adaptation costs tend to decrease due to the increased similarity of retrieved cases. The swamping utility problem occurs when the increased retrieval cost swamps the adaptation cost savings. Recent arguments propose that current computing resources and limited case base sizes for many tasks can make this less important in practice [14]. However, CBR for domains such as big data health care (*e.g.*, [13]) and large-scale e-commerce, for which customer data can measure in the hundreds of millions of cases will continue to face challenges (cf. [15] for an alternative approach to addressing the utility problem, based on big data retrieval methods). Likewise, provenance capture for e-science can result in extremely large provenance cases [4] making case-base compression potentially important for sheer case size. In addition, controlling size can be important even without extreme size, if maintenance requires manual intervention or if the case-bases will be transmitted or replicated, as possible in distributed CBR.

2.2 Knowledge Container Transfer

The relationship of the four CBR knowledge containers—vocabulary, similarity measure, case base, and adaptation knowledge—has given rise to much research on knowledge container transfer, such as improving adaptation knowledge by transfer from cases [10,16,17,28,29,36], and from adaptation knowledge to similarity [18]. Whenever a case-based problem-solver adapts a previous case and stores the result, it can be seen as transferring some of its adaptation knowledge into the case base, in a lazy manner, on demand. ECC, which generates ghost cases by adapting existing cases and adding them to the case base, can be seen as performing eager knowledge transfer from the adaptation component to the case base.

2.3 Exploration of the Problem Space Versus Exploitation of Cases in the Case Base

The notion of exploration versus exploitation concerns how an agent should allocate effort between exploiting existing resources versus exploring in search of others. The explore/exploit trade-off has proven a useful framework in many fields [12]. In traditional case-base maintenance, all maintenance effort is focused on exploitation of existing cases; the ECC process of generating ghost cases using adaptation knowledge explores the space of potential cases. ECC provides both existing cases and the fruits of exploration to CNN to select those cases expected to be most valuable in the compressed case base.

Contrasts Between Ghost Case Generation and Adaptation on Demand: Both ECC and normal CBR do adaptation, but the effect of ECC is different from simply compressing the original case base and adapting the retained cases to solve new problems. For ECC, the system selects new problems to solve. This could potentially be guided by trend detection, to hypothesize areas in which additional case coverage is especially important. For example, the system could note shifts in the types of problems the system is solving [37] to focus ghost case generation there.

When a CBR system adds ghost cases to the case base, it must do so without benefit of the feedback that often enables CBR systems to detect and repair solution flaws. Consequently, the solutions of ghost cases are not guaranteed to be correct. However, because the ghost case is generated in advance, there is an opportunity to seek confirmation of its solution to avoid future failure. (Note that even if the solution proposed by the ghost case remains unverified, and if later application of the ghost case results in an erroneous solution, the same failure would have occurred if the case had been adapted on the fly.)

Benefits of Exploration: Augmenting the case base with ghost cases prior to compression has four primary potential benefits:

1. Potential to improve competence preservation: We hypothesize that for less representative case bases, ECC will enable increased compression for a given competence retention level. We test this hypothesis in Sect. 6. (We expect results to depend on the representativeness of the original case base and on the density of the original case base: If the original case base covers only

a small region but all problems fall within that region, adding ghost cases for problems outside that region might exact a penalty as relevant cases are "crowded out" by the ghost cases.)

2. Potential to improve adaptation efficiency: If ECC applies a performance-based criterion for case retention that reflects not only competence but also adaptation cost [24], an ECC-compressed case base could enable more efficient problem-solving, by adding ghost cases useful for decreasing expected adaptation cost. Where adaptation is not fully reliable but reliability can be estimated by similarity distance, such ghost cases can be chosen to reduce expected average similarity distances, and, consequently, to increase expected solution reliability.
3. Potential to focus active learning/external pre-verification of adaptations: If case adaptation is unreliable, the quality of ghost cases is not guaranteed. However, by selecting useful ghost cases, ECC identifies good candidates for external case acquisition/verification. For example, an expert could be asked to provide the solutions to these problems, or to review system-generated solutions (this may be easier than solution generation, *e.g.*, verifying the routing of pipes in the design of a house is easier than finding a routing).
4. Potential to extend the reach of adaptation: Many CBR systems restrict the adaptation of any problem to a single adaptation step, limiting the problems they can solve (cf. [6]). When ghost cases are generated by applying an adaptation, future adaptations can start from that adapted state, effectively enabling two-step paths for adaptations going through that case. Generating ghost cases from longer adaptation paths further extends the range of problems and decreases adaptation costs.

3 The Expansion-Contraction Compression Algorithm

The ECC algorithm is described in Algorithm 1. Inputs to the algorithm include the maximum length of adaptation paths for generating ghost cases (*ghostSteps*) and the criterion for whether a case can be adapted to solve a given problem (*coverageCriterion*), which we implement as a similarity threshold. ECC first expands the case base by adapting selected cases, according to an adaptation procedure which selects adaptations to perform.

Because ECC performs adaptations in the absence of a specific problem to solve, many strategies are possible for choosing the adaptation, *e.g.*, selecting a random adaptation, selecting a high confidence adaptation, selecting an adaptation expected to produce the greatest difference between old and new solutions, etc. Also, when generating a ghost case, ECC must adjust not only the solution, but also the problem description of the case, to keep the new problem and solution consistent. The adjustment is derived from the problem description part of the chosen adaptation rule—if the rule normally addresses a given difference D between an input problem and a retrieved case, the problem part of the ghost case should be the problem of the current case, adjusted by D.

After generating ghost cases, ECC then compresses the expanded case base by CNN, presenting cases in order of decreasing coverage (other ordering criteria, such as relative coverage [34], could be used as well). If the resulting case base size is below the size limit, ECC "fills out" the additional capacity by adding cases up to the size limit, prioritized by estimated competence contribution.

Algorithm 1. The ECC Algorithm

```
Input:
caseBase: The case base to compress
ghostSteps: the maximum number of adaptation steps for generating ghost cases
    from the case base.
targetSize: the target number of cases in the compressed case base
coverageCriterion: test for whether a case can be adapted to solve another case

Output: Compressed case base

$expandedCaseBase \leftarrow caseBase$
$ghostCases \leftarrow \emptyset$
for $ghostLevel = 0$ to $ghostSteps$ do
    for all $case$ in $expandedCaseBase$ do
        $ghostCases \leftarrow ghostCases \cup adapt(case)$
    end for
    $expandedCaseBase \leftarrow expandedCaseBase \cup ghostCases$
end for
$expandedCaseBase \leftarrow sort(expandedCaseBase, coverage, descending)$
$contractedCaseBase \leftarrow cnn(expandedCaseBase, targetSize, coverageCriterion)$
$additionalCases \leftarrow limit(expandedCaseBase - contractedCaseBase,$
    $targetSize - size(contractedCaseBase))$
$contractedCaseBase \leftarrow contractedCaseBase \cup additionalCases$
return $contractedCaseBase$
```

4 Evaluation

4.1 Experimental Questions

To evaluate expansion-contraction compression, we considered five questions. For the first four, because our tests used standard domains without associated adaptation knowledge, we modeled adaptation based on similarity. Our experiments considered effects of compression on both competence (measured by number of test problems solved) and solution quality (average similarity between problems and the cases retrieved for them):

1. How does ECC affect preservation of quality compared to conventional competence-based compression, for varying levels of representativeness?
2. How does the ECC affect preservation of competence compared to conventional competence-based compression, for varying levels of representativeness?

3. How does the number of steps in the adaptation path to generate ghost cases affect competence retention of ECC, and how does competence compare to CNN?
4. How does sparsity of the initial case base affect the relative preservation of competence of ECC and CNN?

We then tested ECC in a standard domain augmented with automatically generated adaptation rules, to examine:

5. How does ECC affect preservation of competence, quality, and accuracy when applying generated adaptation rules?

5 Experimental Design

Data: The evaluation used five data sets: Houses, with 781 cases and 8 features, from the Datasets Wiki of the California Polytechnic University Computer Science Department [3], and four from the UCI Machine Learning Repository [8]: Iris, with 150 cases and 5 features, Wine, with 178 cases and 14 features, Car Evaluation, with 1,728 cases and 7 features, and Wine Quality, with 1,599 cases and 12 features.

Generating the Case Base and Problem Stream: Each trial partitioned each case base into three random subsets of equal size: (1) training cases (33%), (2) testing cases (33%), and (3) potential ghost cases (33%). Because of the random selection of the subsets, initially the training cases have normal representativeness for the data (the effectiveness of standard competence-based compression suggests that these are reasonably representative). To test the effect of decreased representativeness, one of the experimental conditions modifies the case bases to place a gap in a region of the case base (as might exist, for example, if cases reflected seasonally varying outcomes and no problems had yet been encountered for a particular season). The gap generation process picks a random case as the starting point for the gap, and then removes all of the problems from the training case base within a given similarity threshold (the gap radius). The testing cases and potential ghost cases remain in their original distribution, without an added gap.

Modeling Adaptation and Generating Ghost Cases: None of the data sets include adaptation knowledge. The first four experiments simulate adaptation-driven generation of ghost cases as follows. The experiment repeatedly picks a random case in the training data and filters the set of potential ghost cases for cases within a similarity threshold (the coverage criterion) required for adaptability from the selected training case. Those cases are then treated as the result of an adaptation and stored as ghost cases. This simulates deriving new cases by applying adaptation rules of limited power to the training case base. To test the effects of more powerful adaptation, additional experiments apply this process recursively, with selection of sequences of successive cases to simulate applying chains of one, two, or three adaptation steps.

Generating Adaptation Rules: For the fifth experiment, we applied the case difference heuristic approach [10] to automatically generate a set of adaptation rules from the data. These rules were then used to generate ghost cases by adaptation, rather than drawing on potential ghost cases as in experiments one through four.

Our rule generation approach repeatedly selects two random cases, ascribes the difference in their solutions to the difference in their problems, and forms a rule to apply the same difference to a solution. The rule is applied when the input problem and the problem of the retrieved case have a difference similar to the one from which the rule was generated. Given a problem and retrieved case, the single rule for the most similar difference was applied. For similarity, categorical features in problem descriptions were only considered to match if identical. The rules adjusted the solution values by the proportional difference of the solution values of the cases from which they were generated.

Experimental Procedure: Compression by condensed nearest neighbor [11] was compared to compression by ECC. For comparisons with CNN, we use a version of CNN that, like ECC, "fills out" the case base up to the size limit, so that both methods have access to the same number of cases.

For both ECC and CNN, cases were sorted in descending order of coverage. Compression was done in steps of 10% from 100% to 10% of the size of the uncompressed case base. Each level of compression starts from the uncompressed case base (not the result of the previous level of compression). Each experiment runs for ten trials with different randomly chosen partitions, with results averaged over those runs.

6 Experimental Results

Q1: Relative preservation of quality for different representativeness levels: Figure 1 compares the absolute quality between CNN and ECC strategies for the Houses case base. The coverage criterion for deriving ghost cases and solving testing problems is 5%. Each of the four graphs in Fig. 1 uses a different value for the gap radius. When there is no gap, this value is 0%, a small gap is 5%, a medium gap is 10%, and a large gap is 20%. The size of the gap is measured not in the number of cases that the experiment can remove but in the similarity distance to the most different case that the experiment can remove. The horizontal axis shows the proportionate case base size of the compressed case base, ranging from 100% to 10% in steps of 10%.

In all four graphs of Fig. 1, ECC uses adaptation paths with a maximum length of two steps. The vertical axis shows the quality of the compressed case base. Note that similarity of a retrieved case counts towards the average quality even when the coverage criterion is not met. Quality can fall anywhere in the range from 0 to 1, inclusive. The top and bottom graphs use different scales for the vertical axis (0.91 to 0.97 on the top, and 0.87 to 0.95 on the bottom) in order to "zoom in" on the difference between the strategies.

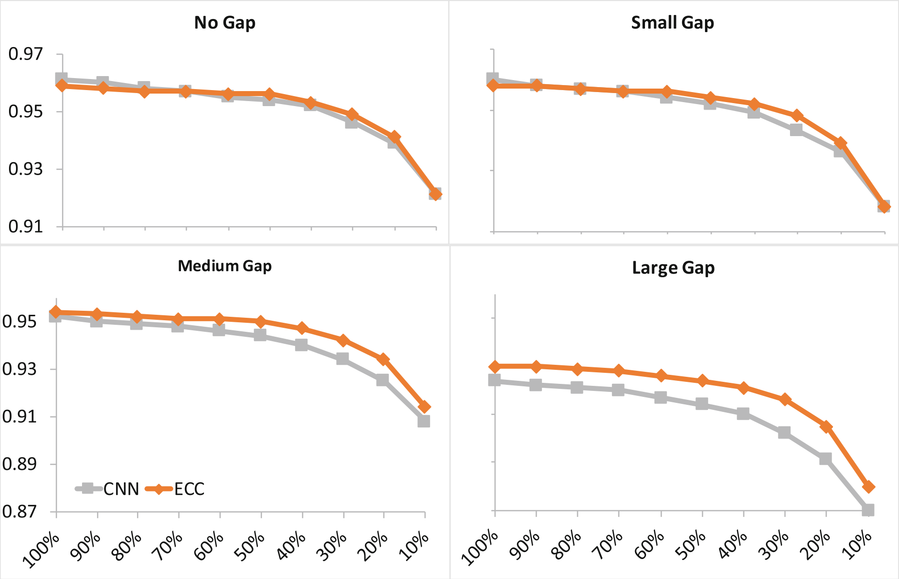

Fig. 1. Absolute quality for condensed nearest neighbor and expansion-contraction compression on House data set, for four different gap sizes in the training data.)

When the training case base has no gap (top left), CNN outperforms ECC from 100% to 80% of the size of the original case base. However, from 70% to 10% size, ECC leads. Because neither the set of training cases nor the set of testing problems has a gap, the training case base is expected to be approximately representative of the testing problems. Therefore, in this graph, we believe that the addition of ghost cases cannot be improving quality by improving representativeness, and must be doing so simply by providing a wider pool of cases from which CNN can select alternatives. This was a small effect, but the benefit of the increased choice is an interesting result that contradicted our initial hypothesis. The Question 2 results show a similar but more dramatic effect on competence.

When the training case base has a small gap (top right), CNN outperforms ECC only at 100% of the size of the original case base. Thereafter, from 90% to 10%, the expansion-contraction strategy leads. For the medium (bottom left) and large gap (bottom right), ECC dominates CNN throughout. Overall, as the size of the gap increases, and the training case base less represents the testing problems, the quality difference increases between CNN and ECC. This suggests that part of the benefit comes from the choice of additional cases for retention (as with the no-gap graph in the top left), and that part of the benefit comes from correction for nonrepresentativeness.

Q2: Relative preservation of competence for different representativeness levels: Figure 2 compares the relative competence between CNN and ECC. The results are based on using the Houses case base and a coverage criterion of 5% both for deriving ghost cases and for solving testing problems. As for Figs. 1 and 2 includes four graphs showing different sizes for the gap in the training case base, and the horizontal axis shows the size of the case base. However, in Fig. 2, the vertical axis shows the relative competence, the ratio of the observed competence to the maximum competence observed with any strategy and any case base size (intuitively, the maximum competence might be expected to be at maximum size, but this need not always hold). The top and bottom graphs use different scales for the vertical axis (60% to 100% on the top, and 20% to 100% on the bottom) in order to "zoom in" on the difference between the strategies.

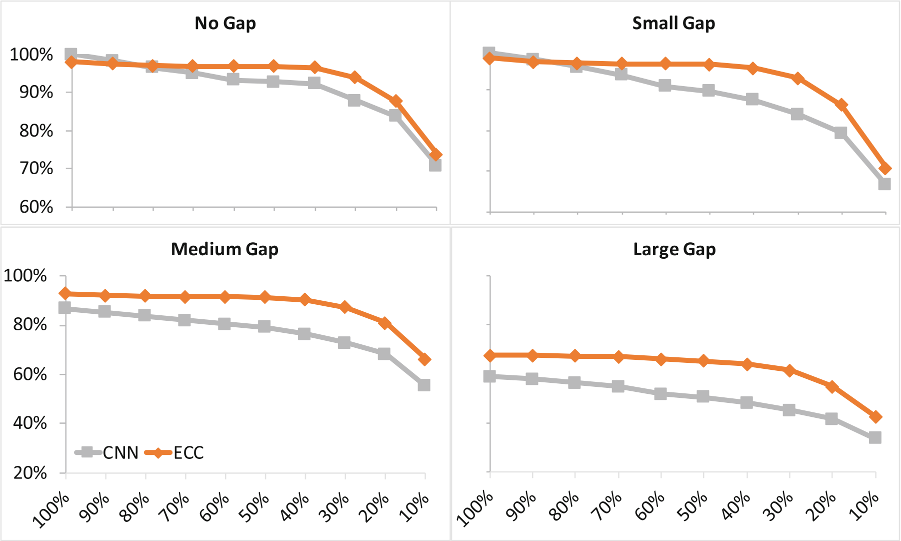

Fig. 2. Relative competence for condensed nearest neighbor and expansion-contraction compression on Houses data set with four different representativeness levels

As shown in the no gap (top left) and small gap (top right) graphs, CNN outperforms ECC at 100% and 90% of the size of the original case base. Thereafter, from 80% to 10%, ECC leads over condensed nearest neighbor. For the medium gap (bottom left) and large gap (bottom right) graphs, ECC dominates CNN throughout all case base sizes. The amount of the difference between the strategies increases as the size of the gap increases.

Question 3: Effect of length of adaptation path for generating ghost cases: Figure 3 presents the relative competence using CNN and ECC for the Houses case base. The gap radius for this figure is 20% (a large gap). The coverage criterion for deriving ghost cases and solving testing problems is 5%, i.e., simulated adaptations can adapt solutions to address problem differences of up to 5%. Therefore, an adaptation path could require up to four steps to cross the gap. The horizontal axis shows the relative size of the case base decreasing from 100% to 10% in steps of 10%. Each of the lines shows a different upper limit to the number of adaptations in the adaptation path to derive ghost cases. No adaptation, which is equivalent to CNN, is the baseline strategy.

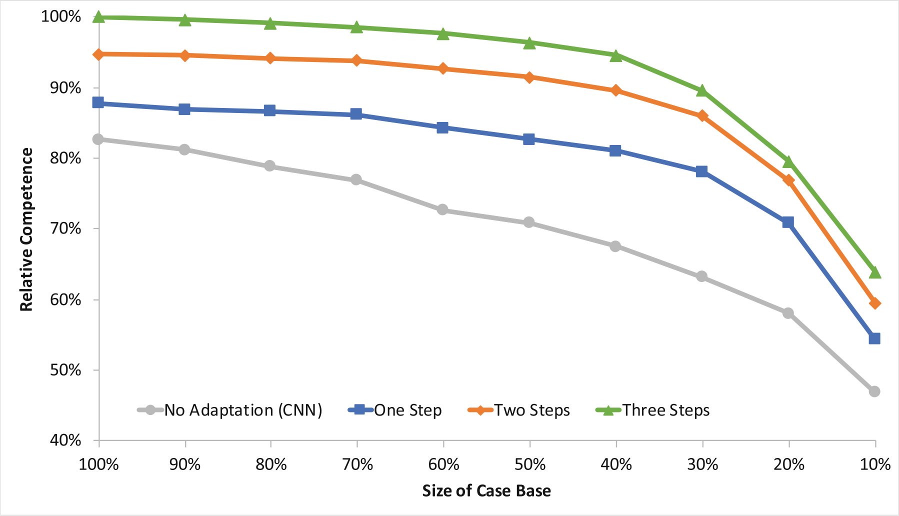

Fig. 3. Relative competence for condensed nearest neighbor and expansion-contraction compression on Houses with three different lengths of adaptation paths

The vertical axis shows the relative competence of the compressed case base. For all adaptation path lengths and case base sizes, ECC dominates CNN. The largest difference in relative competence between CNN and ECC with one step of adaptation is 15% at 30% of the size of the original case base. The smallest difference between these two strategies is 5.1% at 100% size. The relative competence for each adaptation path length varies consistently, with longer paths associated with greater competence retention. The largest difference in relative competence between CNN and ECC with three steps of adaptation is 27% at 40% of the size of the original case base. The smallest difference is 17% at 10% size.

Question 4: Effect of case base sparsity: Figure 4 compares the relative competence between CNN and ECC with a sparse case base; the sparse case base models a case base in the early phases of case base growth. The basic presentation of results follows Fig. 2, but the underlying experiment uses different proportions for the partitions for training, testing, and ghost cases. As described in the experimental design (Sect. 5), the partitions for Figs. 1, 2, and 3 are equal thirds, but for Fig. 4, the proportions are 10% training, 70% testing, and 20% potential ghost cases. The competence trend resembles Fig. 2 but decreases more steeply because the case base begins with fewer cases.

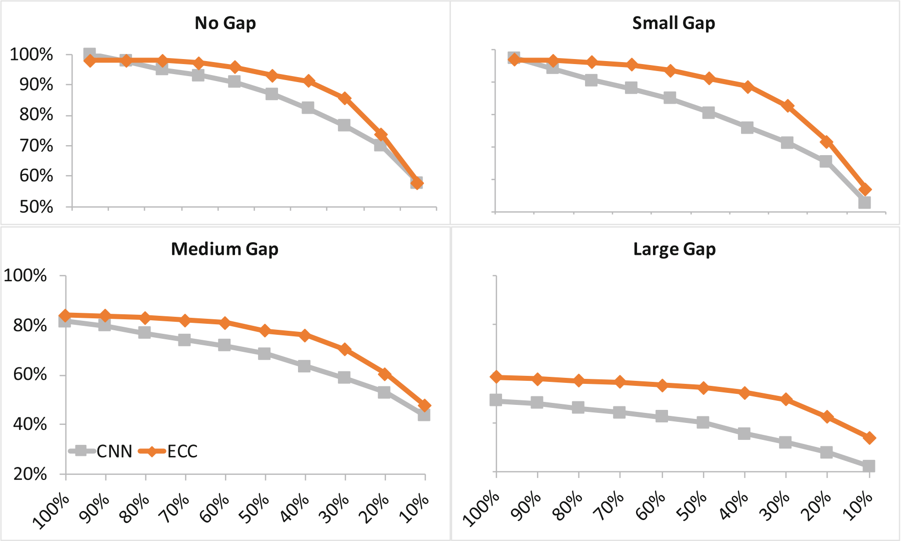

Fig. 4. Relative competence for condensed nearest neighbor and expansion-contraction compression on Houses with a sparse initial case base and four different representativeness levels

Table 1 shows the relative competence difference between CNN and ECC on five different case bases, each modified by adding a medium gap in the training data, making it non-representative. For all five, the gap radius is set to twice the coverage criterion (for House, this corresponds to the lower left graph in Fig. 2). ECC outperformed CNN on four of the five case bases (Houses, Iris, Wine, and Wine Quality), but not on Car Evaluation. This suggests that the expansion-contraction strategy might benefit compression across many domains and raises the question of which factors determine whether it will be beneficial.

Table 1. Difference in relative competence between ECC and CNN on five different case bases

Case base	No compression	33% compression	67% compression
Houses	6.6%	11.2%	15.8%
Iris	1.2%	7.6%	7.0%
Wine	2.8%	9.6%	13.4%
Car evaluation	−3.5%	0.0%	−8.0%
Wine quality (Red)	9.4%	13.4%	14.9%

Question 5: Effect of ECC on competence, quality, and accuracy for sample adaptation rule sets:

Experiments with adaptation rules used the Houses data set, with a gap radius of 20% (corresponding to a large gap in the previous experiments), a 5% coverage criterion, and 40 automatically-generated rules for each trial. ECC used the adaptation rules both to generate ghost cases and to adapt solutions. Figure 5 shows the average preservation of competence and quality during compression by ECC and CNN. ECC outperformed CNN in both dimensions at all compression levels. Thus ECC was beneficial both for modeled adaptation and adaptation with generated adaptation rules.

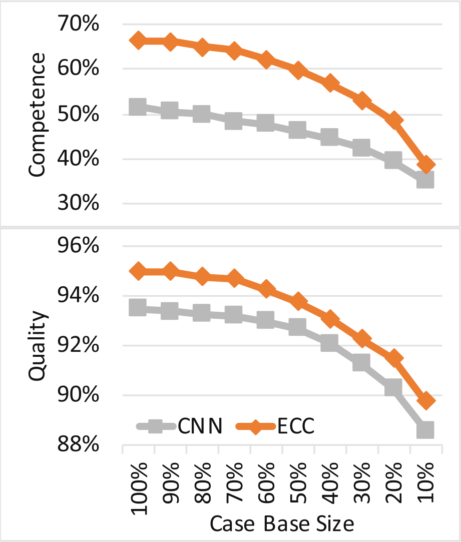

Fig. 5. Competence and quality for ECC and CNN on Houses data with adaptation rules.

7 Future Work

In our experiments, ghost case generation is an unguided process, generating ghost cases from randomly-selected cases using randomly-generated adaptations. In addition, ghost cases are generated within neighborhoods of existing cases, making ghost cases most likely to be added near regions that are already densely populated. Ghost cases can be useful there, for example, as spanning cases [33], bridging two competence regions. However, such placement may not help to populate a distant sparse region. Identifying and targeting sparse regions on which to focus ghost case generation might increase the benefit of ECC. On the other hand, populating such regions might be detrimental if representativeness generally holds. Thus determining guidance strategies for ghost case generation and assessing their effects is an interesting area for future study.

For example, areas to target for ghost generation could be selected by dividing the case base into regions and applying Monte Carlo methods to assess case base density [25]. Another interesting direction would be to develop maintenance strategies that tracked, reasoned about, and responded to expected future case distributions, to explicitly determine the expected utility of exploring areas of the unseen case space.

8 Conclusion

This paper proposed a new approach to case-base compression, Expansion-Contraction Compression (ECC), aimed at achieving better competence retention when the representativeness assumption may not hold. The ECC approach contrasts with previous approaches, which focus on how best to exploit the cases in the case base, in going outside problems addressed by the case base to explore the larger problem space as licensed by case adaptation knowledge. ECC can be seen as applying a knowledge container transfer strategy: It uses adaptation knowledge to generate new case knowledge, ghost cases, which are then added to the pool of cases available to the compression algorithm. Chosen ghost cases can be retained as-is or can be used to guide verification or active learning of new cases. Experiments support the expected result of ECC improving competence retention for less-representative case bases. Surprisingly, ECC also often improved competence retention even for standard case bases. Thus the ECC approach appears promising. Interesting questions remain for studying factors affecting ECC performance and guiding ghost case generation to maximize the effectiveness of ECC.

References

1. Angiulli, F.: Fast condensed nearest neighbor rule. In: Proceedings of the Twenty-second International Conference on Machine Learning, pp. 25–32. ACM, New York (2005)
2. Brighton, H., Mellish, C.: Identifying competence-critical instances for instance-based learners. In: Liu, H., Motoda, H. (eds.) Instance Selection and Construction for Data Mining, vol. 608, pp. 77–94. Springer, Berlin (2001). https://doi.org/10.1007/978-1-4757-3359-4_5
3. Houses Data Set, May 2009. https://wiki.csc.calpoly.edu/datasets/wiki/Houses
4. Cheah, Y.-W., Plale, B., Kendall-Morwick, J., Leake, D., Ramakrishnan, L.: A noisy 10GB provenance database. In: Daniel, F., Barkaoui, K., Dustdar, S. (eds.) BPM 2011. LNBIP, vol. 100, pp. 370–381. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28115-0_35
5. Cunningham, P., Nowlan, N., Delany, S., Haahr, M.: A case-based approach to spam filtering that can track concept drift. Technical report. TCD-CS-2003-16, Computer Science Department, Trinity College Dublin (2003)

6. D'Aquin, M., Lieber, J., Napoli, A.: Adaptation knowledge acquisition: a case study for case-based decision support in oncology. Comput. Intell. **22**(3/4), 161–176 (2006)
7. Delany, S.J., Cunningham, P.: An analysis of case-base editing in a spam filtering system. In: Funk, P., González Calero, P.A. (eds.) ECCBR 2004. LNCS, vol. 3155, pp. 128–141. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-28631-8_11
8. Dheeru, D., Karra Taniskidou, E.: UCI Machine Learning Repository (2017). http://archive.ics.uci.edu/ml
9. Francis, A.G., Ram, A.: A comparative utility analysis of case-based reasoning and control-rule learning systems. In: Lavrac, N., Wrobel, S. (eds.) ECML 1995. LNCS, vol. 912, pp. 138–150. Springer, Heidelberg (1995). https://doi.org/10.1007/3-540-59286-5_54
10. Hanney, K., Keane, M.T.: Learning adaptation rules from a case-base. In: Smith, I., Faltings, B. (eds.) EWCBR 1996. LNCS, vol. 1168, pp. 179–192. Springer, Heidelberg (1996). https://doi.org/10.1007/BFb0020610
11. Hart, P.E.: The condensed nearest neighbor rule. IEEE Trans. Inf. Theory **14**, 515–516 (1968)
12. Hills, T., Todd, P., Lazer, D., Redish, A., Couzin, I.: Exploration versus exploitation in space, mind, and society. Trends Cogn. Sci. **19**(1), 46–54 (2015)
13. Hoover, W.: Transforming health care through big data. Technical report, Institute for Health Technology Transformation (2013)
14. Houeland, T.G., Aamodt, A.: The Utility problem for lazy learners - towards a non-eager approach. In: Bichindaritz, I., Montani, S. (eds.) ICCBR 2010. LNCS, vol. 6176, pp. 141–155. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14274-1_12
15. Jalali, V., Leake, D.: Harnessing hundreds of millions of cases: case-based prediction at industrial scale. In: Ram, A., Wiratunga, N. (eds.) Case-Based Reasoning Research and Development, ICCBR 2018, vol. 6880. Springer, Berlin (2018). https://doi.org/10.1007/978-3-642-23291-6
16. Jalali, V., Leake, D., Forouzandehmehr, N.: Ensemble of adaptations for classification: learning adaptation rules for categorical features. In: Goel, A., Díaz-Agudo, M.B., Roth-Berghofer, T. (eds.) ICCBR 2016. LNCS, vol. 9969, pp. 186–202. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-47096-2_13
17. Jalali, V., Leake, D.: Extending case adaptation with automatically-generated ensembles of adaptation rules. In: Delany, S.J., Ontañón, S. (eds.) ICCBR 2013. LNCS, vol. 7969, pp. 188–202. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39056-2_14
18. Leake, D., Kinley, A., Wilson, D.: Linking adaptation and similarity learning. In: Proceedings of the Eighteenth Annual Conference of the Cognitive Science Society, pp. 591–596. Lawrence Erlbaum, Mahwah (1996)
19. Leake, D., Schack, B.: Flexible feature deletion: compacting case bases by selectively compressing case contents. In: Hüllermeier, E., Minor, M. (eds.) ICCBR 2015. LNCS, vol. 9343, pp. 212–227. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24586-7_15

20. Leake, D., Schack, B.: Adaptation-guided feature deletion: testing recoverability to guide case compression. In: Goel, A., Díaz-Agudo, M.B., Roth-Berghofer, T. (eds.) ICCBR 2016. LNCS, vol. 9969, pp. 234–248. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-47096-2_16
21. Leake, D., Sooriamurthi, R.: Managing multiple case-bases: dimensions and issues. In: Proceedings of the Fifteenth International Florida Artificial Intelligence Research Society Conference, pp. 106–110. AAAI Press, Menlo Park (2002)
22. Leake, D., Whitehead, M.: Case provenance: the value of remembering case sources. In: Weber, R.O., Richter, M.M. (eds.) ICCBR 2007. LNCS, vol. 4626, pp. 194–208. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74141-1_14
23. Leake, D.B., Wilson, D.C.: When experience is wrong: examining CBR for changing tasks and environments. In: Althoff, K.-D., Bergmann, R., Branting, L.K. (eds.) ICCBR 1999. LNCS, vol. 1650, pp. 218–232. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48508-2_16
24. Leake, D.B., Wilson, D.C.: Remembering why to remember: performance-guided case-base maintenance. In: Blanzieri, E., Portinale, L. (eds.) EWCBR 2000. LNCS, vol. 1898, pp. 161–172. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-44527-7_15
25. Leake, D., Wilson, M.: How many cases do you need? Assessing and predicting case-base coverage. In: Ram, A., Wiratunga, N. (eds.) ICCBR 2011. LNCS, vol. 6880, pp. 92–106. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-23291-6_9
26. Lu, N., Zhang, G., Lu, J.: Concept drift detection via competence models. Artif. Intell. **209**, 11–28 (2014)
27. Mathew, D., Chakraborti, S.: Competence guided model for casebase maintenance. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, pp. 4904–4908. International Joint Conferences on Artificial Intelligence (2017)
28. McDonnell, N., Cunningham, P.: A Knowledge-light approach to regression using case-based reasoning. In: Roth-Berghofer, T.R., Göker, M.H., Güvenir, H.A. (eds.) ECCBR 2006. LNCS, vol. 4106, pp. 91–105. Springer, Heidelberg (2006). https://doi.org/10.1007/11805816_9
29. McSherry, D.: An adaptation heuristic for case-based estimation. In: Smyth, B., Cunningham, P. (eds.) EWCBR 1998. LNCS, vol. 1488, pp. 184–195. Springer, Heidelberg (1998). https://doi.org/10.1007/BFb0056332
30. Plaza, E., McGinty, L.: Distributed case-based reasoning. Knowl. Eng. Rev. **20**(3), 315–320 (2005)
31. Richter, M.: Knowledge Containers (2005). http://pages.cpsc.ucalgary.ca/mrichter/Papers/Knowledge%20Containers.pdf
32. Smyth, B., Cunningham, P.: The utility problem analysed. In: Smith, I., Faltings, B. (eds.) EWCBR 1996. LNCS, vol. 1168, pp. 392–399. Springer, Heidelberg (1996). https://doi.org/10.1007/BFb0020625
33. Smyth, B., Keane, M.: 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, pp. 377–382. Morgan Kaufmann, San Mateo (1995)

34. Smyt, B., McKenna, E.: Footprint-based retrieval. In: Althoff, K.-D., Bergmann, R., Branting, L.K. (eds.) ICCBR 1999. LNCS, vol. 1650, pp. 343–357. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48508-2_25
35. van Someren, M., Surma, J., Torasso, P.: 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, pp. 477–488. Springer, Berlin (1997)
36. Wilke, W., Vollrath, I., Bergmann, R.: Using knowledge containers to model a framework for learning adaptation knowledge. In: ECML Workshop Notes. LIS, Faculty of Informatics and Statistics (1998)
37. Wilson, D., Leake, D.: Maintaining case-based reasoners: dimensions and directions. Comput. Intell. **17**(2), 196–213 (2001)
38. Wilson, D., Martinez, T.: Reduction techniques for instance-based learning algorithms. Mach. Learn. **38**(3), 257–286 (2000)