Adaptation-Guided Case Base Maintenance

Vahid Jalali and **David Leake**
School of Informatics and Computing, Indiana University
Informatics West, 901 E. 10th Street
Bloomington, Indiana 47408, USA

Abstract

In case-based reasoning (CBR), problems are solved by retrieving prior cases and adapting their solutions to fit; learning occurs as new cases are stored. Controlling the growth of the case base is a fundamental problem, and research on *case-base maintenance* has developed methods for compacting case bases while maintaining system competence, primarily by competence-based deletion strategies assuming static case adaptation knowledge. This paper proposes *adaptation-guided case-base maintenance* (AGCBM), a case-base maintenance approach exploiting the ability to dynamically generate new adaptation knowledge from cases. In AGCBM, case retention decisions are based both on cases' value as base cases for solving problems and on their value for generating new adaptation rules. The paper illustrates the method for numerical prediction tasks (case-based regression) in which adaptation rules are generated automatically using the *case difference heuristic*. In comparisons of AGCBM to five alternative methods in four domains, for varying case base densities, AGCBM outperformed the alternatives in all domains, with greatest benefit at high compression.

Introduction

Case-based reasoning (CBR) (e.g., Lopez de Mantaras *et al.*, 2005) is a problem solving paradigm in which new problems are solved by retrieving solutions of similar previous problems and adapting them based on problem differences. Case adaptation is commonly performed by *adaptation rules*. After solving a new problem, the CBR system learns by storing the new case in the case base. If all cases are retained, retrieval cost increases may impair problem-solving speed (e.g., Smyth and Cunningham, 1996). Consequently, much CBR research has focused on case-base maintenance, and especially on how to select which cases to retain (for a sampling, see Leake et al., 2001). Much of this work focuses on competence-preserving case deletion, which aims to delete cases whose loss is expected to be least harmful to overall competence (Smyth and Keane 1995).

Case-base maintenance research has long recognized that the competence contributions of cases depend on the knowledge available to adapt them, taking static adaptation knowledge into account as an independent factor. However, case and adaptation knowledge may not be independent. Methods have been developed for generating adaptation knowledge from the cases in the case base (e.g., Hanney and Keane, 1996), and have been applied to just-in-time generation of case adaptation rules, based on the contents of the case base (Jalali and Leake 2013b). When adaptation knowledge is generated from cases, an interesting question is whether competence-based maintenance can be improved by basing it both on cases' potential direct contribution to problem-solving and to their potential contribution to case adaptation knowledge. This paper introduces a new case base maintenance approach, *Adaptation-Guided Case Base Maintenance* (AGCBM). whose key principle is simultaneously to consider both roles. Coupling case selection to adaptation contributions is in the spirit of *adaptation-guided retrieval*, which couples retrieval decisions to case adaptability (Smyth and Keane 1998).

This paper first reviews prior approaches to condensing case sets. It then describes AGCBM and illustrates its application to CBR for regression (numerical prediction) tasks. It presents a specific instantiation of the AGCBM approach, AGCBM1, and evaluates it compared both to standard case base maintenance methods and to two ablated versions of AGCBM1 designed to test AGCM1's case ranking, in four standard domains. The evaluation shows encouraging results for accuracy compared to alternative compression methods, especially for case bases with low density compared to the problem space.

Related Work

The foundation of much work on compressing example sets is Hart's (1968) condensed nearest neighbor (CNN) method, which selects reduced sets of cases to be applied by the k-nearest-neighbor (k-NN) algorithm. CNN iteratively selects a subset of a training set to retain, starting from the empty set and adding cases that are not correctly solved by previously-selected cases. The process is repeated, starting from the currently selected set, until all cases are correctly solved or a size limit is reached. Methods built on refinements of CNN framework include Aha et al.'s (1991) IB2, which starts from an empty training set and adds misclassified instances; their IB3 addresses the problem of keeping noisy instances by only retaining an instance if a lower bound on its contribution to accuracy is significantly higher than its class frequency.

Because the results of CNN depend on the order in which cases are considered, a common research focus has been on how to order cases to present to CNN (Angiulli 2005; Wilson and Martinez 2000; Brighton and Mellish 2001; Delany and Cunningham 2004). Smyth (1995; 1998) proposes the use of *relative coverage*. The coverage of a case is the set of target problems that can be solved by that case; relative coverage considers how many other cases in the case base can solve the cases in the coverage set, favoring cases which cover cases reachable by fewer other cases. Such methods are referred to as *footprint-based* methods.

Craw, Massie, and Wiratunga (2007) extend footprint based maintenance by focusing on interaction at class boundaries, introducing local case complexity based on the number of cases in the neighborhood of a given case that agree/disagree with its class label, to discard cases with extremely low complexity (redundant cases) or high complexity (noisy cases). Lieber (1995) proposes aiming case deletion at maintaining maximal case diversity. Racine and Yang (1997) propose deletion based on case subsumption, first deleting cases subsumed by other cases. Brodley's (1993) Model Class Selection (MCS) method for k-NN compares the number of times an instance appears among the top k nearest neighbors of other instances while its class label matches or disagrees with the label of those cases. If the number of disagreements for an instance is higher than the number matched, the case is discarded, otherwise it is retained. Zhang (1992) proposes retaining instances closer to the center of clusters, rather than instances close to the borders. Some ordering methods for CNN introduce considerations beyond competence. Leake and Wilson (2000) propose *relative performance*, aimed at reducing case adaptation cost by considering the expected contribution of a case to the system's adaptation performance.

The central contribution of adaptation-guided case-base maintenance is to go beyond treating case adaptation knowledge as fixed when doing maintenance, instead guiding retention decisions according to coordinated consideration of the value of the case as a source case, and as data for generating case adaptation knowledge.

Adaptation-Guided Case Base Maintenance

Cases may be used both as the source cases to adapt to solve new problems, and as the basis for generating adaptation knowledge. Hanney and Keane's (1996) *case difference heuristic* approach generates adaptation rules by comparing two cases (the "composing cases" of the adaptation rule), noting the differences between the two problems and the two solutions. Given a new problem and a retrieved source case with a similar difference, the resulting rule will adjust the source case's solution by the previously observed difference.

Exhaustive generation of a set of adaptation rules in advance could result in an overwhelming number of rules, leading to a rule maintenance problem. However, methods such as the case difference heuristic can also be used for lazy generation of adaptation rules as needed (Jalali and Leake 2013b). This raises the question, addressed in this paper, of how to guide case-base maintenance when cases are used both for direct problem-solving and for generating adaptations. The two roles are coupled, because the competence of a source case depends on the availability of adaptations to adjust its value, which in turn depends on the pairs of cases available to generate adaptation rules.

Adaptation-Guided Case Base Maintenance (AGCBM) ranks the competence contributions of cases according both to their contributions as source cases for problem-solving and their contribution to building adaptations to be used to adjust the values of source cases. It then applies CNN to the case base, in order of decreasing contribution.

Calculating Case Contributions

Applying the AGCBM principle requires methods for calculating two values for each case: (1) its contribution to competence when it is used as a source case, and (2) its contribution to competence when used to generate adaptation rules. The following section describes how we have addressed (1) and (2) in a method for case-based regression, which we refer to as AGCBM1, which is tested in our evaluation.

Source Case Contribution AGCBM calculates the *source case contribution* by using the entire initial case base as a training set, with leave-one-out testing using the system's standard CBR process, summing the errors in the values generated. Each leave-one-out test generates solutions based on cases in a neighborhood of the problem. To adapt each case, AGCBM generates adaptation rules using the same process to be used later by the CBR system to process input queries. Blame for erroneous values is assigned to each of the cases in the neighborhood used to calculate the value, as well as to each of the adaptation rules applied to the cases.

Cases never used as source cases in the leave-one-out test process are assigned contributions of zero. For any case C, used $N > 0$ times as a source case for solving the problems in the test set, let $EstErr_i(C)$ designate the error when C is used to solve the i^{th} problem for which it is used. We define $K : [0, \infty) \rightarrow \Re^+$ as, for some small positive ϵ, $K(x) \equiv x + \epsilon$. This determines a non-zero minimum value for perfect predictions. We then define the case knowledge competence of C as:

$$CaseComp(C) \equiv \sum_{i}^{N} \frac{1}{K(EstErr_i(C))} \qquad (1)$$

We note that other formulations of K could be used to tune the effect of particular magnitudes of errors on the competence values (e.g, if errors below a given threshold should be given negligible weight).

Contribution to Adaptation Knowledge AGCBM calculates the *adaptation knowledge contribution* for a case by summing the estimation errors for all predictions in which that case is used as one of the two composing cases for generating an adaptation rule by the case difference heuristic, in the leave-one-out tests. Cases never used to generate adaptations in the leave-one-out tests are assigned contributions of zero.

For a case C, if M is the number of adaptations applied to the training data for which C is a composing case, $EstErr_j(C)$'s represent estimation errors for predictions in which adaptations derived from C , the adaptation knowledge competence $AdaptComp$ of C is calculated as follows (here both $CaseComp$ and $AdaptComp$ use the same function K, but in principle these need not be identical):

$$AdaptComp(C) \equiv \sum_{j=1}^{M} \frac{1}{K(EstErr_j)} \tag{2}$$

Overall Contribution The overall contribution combines the values for contributions to source case and adaptation knowledge. Given a parameter $\alpha \in [0, 1]$ to tune the balance, the overall contribution of C denoted by $Comp(C)$ is calculated as:

$$\begin{aligned} Comp(C) \equiv& \alpha \times CaseComp(C) + \\ & (1 - \alpha) \times AdaptComp(C) \end{aligned} \tag{3}$$

The parameter α can be set to maximize performance based on the training data.

Ranking by Overall Contribution In AGCBM, the case base is initialized with a small number of cases (the number is user-set), which it selects as those closest to the centroids resulting from unsupervised clustering of the case base (AGCBM1 uses k-means clustering). Next, it ranks cases based on their competence values calculated by Eq. 3 and presents them to CNN in order of decreasing score to form the final condensed case base. Alg. 1 summarizes the AGCBM algorithm, where $Next$, $AGCBMRanking$,and $FindCentroids$ respectively denote functions for accessing the next element of an ordered list, for ordering cases based on their competence calculated by Eq. 3, and for finding cases closest to the centroids of the case base. *Value(c)* is the solution value associated with case *c*, and *FindSol(c)* is the CBR procedure used to generate the solution.

Underlying Approach for AGCBM1

Any application of AGCBM depends on (1) an underlying approach for generating adaptation rules from cases, and (2) an underlying approach for generating values from source cases. We tested an approach, AGCBM1, which generates adaptation rules using a simple version of the case difference heuristic (Hanney 1997), in which the problem descriptor and proposed solution adjustment parts of rules are generated by subtracting the input features and solution values of the composing cases of that adaptation rule.

In AGCBM1, *FindSol* and *AGCBMRanking* from Alg. 1 use CAAR (Context-Aware Adaptation Retrieval) (Jalali and Leake 2013a) for estimating case values. CAAR is a case-based regression method designed to use ensembles of adaptation rules, making it well-suited to application with automatically-generated adaptation rules, because the use of ensembles of rules helps to compensate for variations in

Algorithm 1 AGCBM's basic algorithm

Input:
n: number of cases to maintain
CB: case base
Output: a subset of CB consisting of n cases

```
RankedCases ← AGCBMRanking(CB)
CondensedCB ← FindCentroids(CB)
while size(CondensedCB) < n do
   c ← Next(RankedCases)
   EstimationError ← Abs(Value(c) - FindSol(c, CB))
   if EstimationError < threshold then
       Add(CondensedCB, c)
   end if
end while
return CondensedCB
```

Algorithm 2 Algorithm for value estimation (CAAR)

Input:
Q: input query
n: number of source cases to adapt to solve query
r: number of rules to be applied per source case
CB: case base
Output: Estimated solution value for Q

```
CasesToAdapt ← NeighborhoodSelection(Q,n,CB)
NewRules: ←
    RuleGenerationStrategy(Q,CasesToAdapt,CB)
for c in CasesToAdapt do
      RankedRules ← RankRules(NewRules,c,Q)
      ValEstimate(c) ←
      CombineAdaptations(RankedRules, c, r)
end for
return CombineVals(∪_{c∈CasesToAdapt} ValEstimate(c))
```

the quality of the automatically-generated rules (Jalali and Leake 2013b). *FindSol* finds the optimal tunings for CAAR (for the number of source cases to adapt and number of adaptations to be applied per source case) and invokes CAAR by passing it a set of input parameters.

Alg. 2 summarizes CAAR's algorithm. CAAR generates adaptation rules by comparing every case in the case base with its top k nearest neighbors. To estimate the value of an input problem, its closest neighbors are retrieved as source cases, and their values are adjusted by applying an ensemble of adaptations whose results are averaged.

Independence and Ranking: The AGCBM1 ranking function treats the inclusion of each case independently from the others. However, cases' contributions are interrelated—the value of a source case depends on the cases available to generate adaptations for it, and the value of an adaptation depends on the source cases to which it can apply. Yet even with its independence assumption, AGCBM1 provides substantial benefits, which we hypothesize is because the test data set used by AGCBM1 tends to remain representative.

Time Complexity of AGCBM1: AGCBM1s time complexity is determined by the leave-one-out testing process used for case ranking, and by CAAR which is applied by each leave-one-out iteration. CAAR's time complexity is linear in the number of cases in the case base, and tests suggest that good results can be obtained when CAAR considers small numbers of cases (5-10 for the domains tested in (Jalali and Leake 2013a)). The time complexity of the leave-one-out testing is quadratic in the number of cases. To increase efficiency, exhaustive leave-one-out could be replaced or enhanced by other methods (e.g. sampling methods) for scalability. However, we note that compressing the case base is not a routine process. Depending on the application, it might be sufficient to perform compression only once for the life of the system.

Experiments

Experimental Design: We evaluated AGCBM1's performance on four sample domains from the UCI repository (Frank and Asuncion 2010): Automobile (Auto), Auto MPG (MPG), Housing, and Computer Hardware (Hardware). For all data sets, records with unknown values were removed. Only numeric features were used, and similarity was assessed by Euclidean distance. For each feature, values were standardized by subtracting that feature's mean value from the feature value and dividing the result by the standard deviation of that feature. Experiments measure the ability of different methods to maintain accuracy (measured by mean absolute error, MAE) at different compression rates. For Automobile, MPG, Housing and Computer Hardware domains the respective values to estimate are price, mpg, MEDV (median value of owner-occupied homes in $1000's) and PRP (published relative performance). Ten fold cross validation is used for all experiments, and all methods' parameters are tuned by using hill climbing on the training data. The parameters of all methods were tuned by hill-climbing. The parameters to tune for AGCBM1 are the number of source cases to use and the number of adaptations to apply per source case for estimating the case values, and α from Eq. 3. The size limit used in the CNN process was set based on the desired reduction in the training set size.

Experiments compare AGCBM1 to five case base maintenance methods standard in the current literature: Random Deletion (Markovitch and Scott 1993) (Random), CNN (Hart 1968), Coverage-Based Maintenance (Cov) (Smyth and Keane 1995), Reachability-Based Maintenance (Reach) (Smyth and Keane 1995), and Relative Coverage-Based Maintenance (RelCov) (Smyth and McKenna 1999). We also compared two ablated versions of AGCBM1 to test the effect of AGCBM1's case ranking method, as explained in the next section.

We note that prior research on coverage, reachability and relative coverage applied those approaches to classification tasks; we adapted them for regression. The "coverage" of a case was defined as the number of cases for which both (1) it is among their top k nearest neighbors, and (2) the cases's value is within a pre-defined threshold T of the correct value. Analogously the reachability of a case is the number of its top k nearest neighbors whose values differ from it by less than a pre-set threshold.

We also note that the comparison methods focus primarily on retrieval, rather than on retrieval plus more sophisticated adaptation. To the best of our knowledge, none of the existing case-based regression methods using both retrieval and adaptation has been used for case base maintenance, so no exact comparison points are available in the literature. However, if they were applied under the same independence assumption introduced in AGCBM1 (that the case and adaptation contributions can be assessed separately), they would correspond to weakened versions of AGCBM1, in that tests of CAAR (used in AGCBM1) show its results to be more accurate than other retrieval plus adaptation methods (Jalali and Leake 2013a).

Experimental Results: Our experiments address three main questions:

1. How does accuracy of compressed case bases generated by AGCBM1 compare to the accuracy of other candidate methods, for different compression rates?
2. How does AGCBM1's case ranking approach affect performance compared to random ranking?
3. How does a ranking assigning equal weightings to the role of case and adaptation knowledge compare to a ranking in which the balance (determined by α) is tuned using the training data?

Comparison of AGCBM1 to other candidate methods To address question 1, we ran tests comparing accuracy resulting from AGCBM1 to five other methods. Figure 1 shows MAE of all methods, in four domains, for target case base sizes running from a small fraction of the training data to using all available training data. Due to the use of 10-fold cross validation, the maximal size is 90% of the total training set.

In all domains except the Auto domain, AGCBM1 outperforms the other methods. For the Auto domain, for three case base sizes, relative coverage slightly outperforms AGCBM1, by 4%, 1%, and 0.5% when the number of maintained cases is equal to 100, 110 and 130 respectively. In general, Random Deletion shows the worst performance. However, for smaller case base sizes for different domains the worst performance is sometimes a footprint based method. For example, for the Auto domain with minimum case base size, Relative Coverage shows the worst performance while for MPG, Housing and Hardware domains Reachability, Coverage and Random Deletion methods show the worst performance respectively.

We note that the gap between AGCBM1's accuracy and other methods' is biggest for smaller case base sizes; the advantage of AGCBM1 shrinks as the target case base size grows. We expect differences to be most pronounced at high compression, because at low compression, larger numbers of cases provide additional cases which can compensate for suboptimal choices by any method. However, while the performance of all methods becomes closer as compression de-

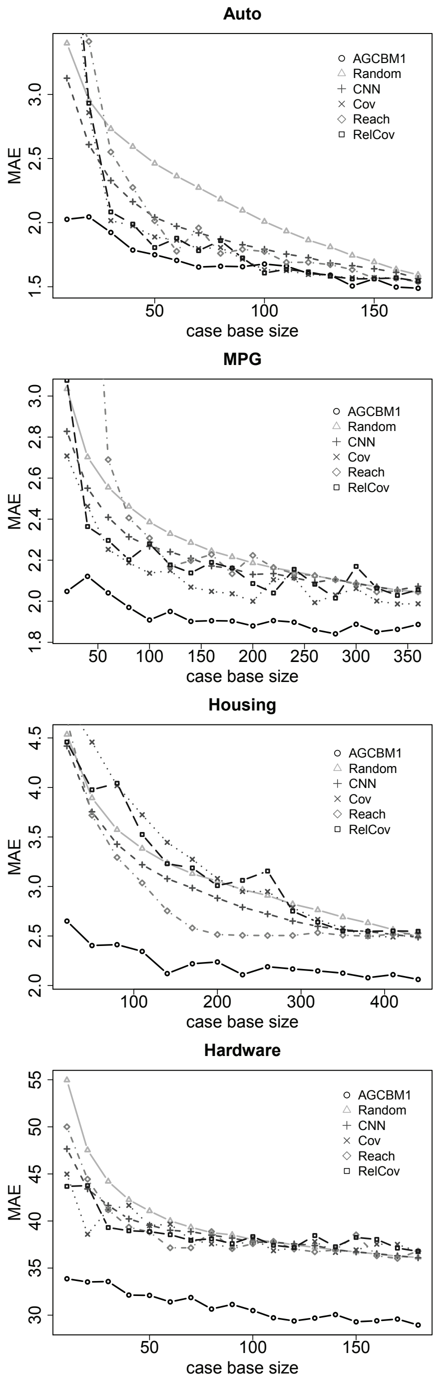

Figure 1: MAE of the candidate methods on four sample domains

creases, AGCBM1 tends to maintain an advantage even for large case bases.

Fig. 2 compares the percent improvement of AGCBM1 and the second-best method, for each domain (always a footprint-based method; Coverage for Auto and MPG domains, Reachability for Housing, and Relative coverage for Hardware) to random deletion. In all domains the greatest gain over Random Deletion is for maximal compression, where AGCBM1 shows 32% to almost 50% improvement over Random Deletion, and the benefit decreases as the case base grows. On the other hand, the trend for the footprint-based methods is not consistent across different domains. In all test domains except Hardware, the best performance for footprint-based methods is observed when 33% of cases in the case base are maintained.

In all domains, when almost the full training set is retained in the case base, the footprint-based method's performance is very close to that of Random Deletion. This is reasonable because all those methods rely on k-NN for estimating target values, and given the same case bases performance would be expected to be almost identical.

Fig. 1 also shows that the gap between the performance of AGCBM1 at the smallest and biggest case base size is smaller than for the other methods. For example, in Auto, MPG, Housing, and Hardware domains increasing the number of maintained cases from the minimum to the full training set size results in 27%, 8%, 11%, and 14% improvement in MAE respectively while these values for Random Deletion method in the same order are 53%, 33%, 45%, and 34%.

Assessing the performance of AGCBM1's case ranking method: Interestingly, for all domains, hill climbing assigned noticeably higher weights to the role of cases for generating adaptations than for the role of cases as source cases (i.e. small α values). To investigate the impact of the weighting on AGCBM1's performance, AGCBM1 was compared to two alternative versions, one which randomly selected cases to retain, and another which used a balanced ($\alpha = 0.5$) weighting for tuning the role of source case versus adaptation contributions compared to the optimal tunings, for which α ranges between 0.2 to 0.05 for different domains.

Fig. 3 illustrates the performance of AGCBM1 and the two variants (Random and Non-optimal weighting) in two of the four sample domains. (The other domains are omitted for reasons of space; performance was similar.) AGCBM1 almost always outperforms the alternatives, but as the number of cases approaches the full training set size, all three methods show almost identical performance.

Statistical Significance A one side paired t-test with 95% confidence interval was used to assess the statistical significance of results achieved by AGCBM1. AGCBM1 was compared to Random Deletion, the best performing footprint-based method, AGCBM1 with random case selection (Ablated1), and AGCBM1 with non-optimal weighting of case and adaptation knowledge (Ablated2). The statistical significances are only reported for the minimum and maximum compressed case base sizes of Figure 1. The null hypothesis is MAE of AGCBM1 being greater than the other methods. Table 1 shows these results.

AGCBM1's performance edge over standard methods is always significant for the maximum compression, and even its variants' performance is significantly better, with the exception in MPG domain for one of the versions. When the compression level is minimal, AGCBM1's performance is only significantly better than the variants for the Housing

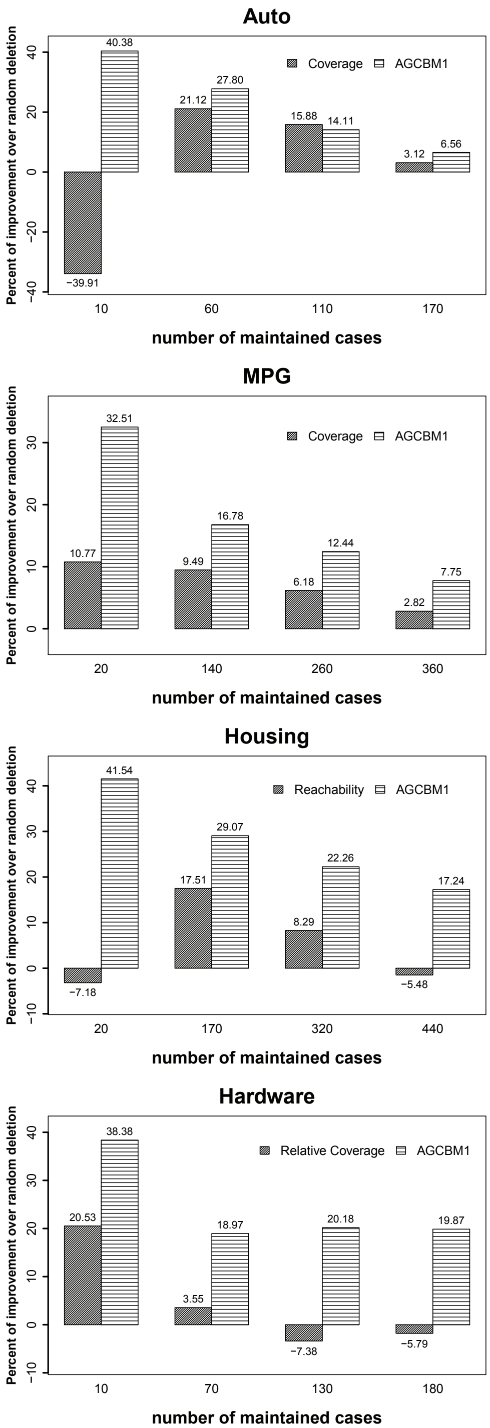

Figure 2: Percent of improvement in MAE over random method at four different case base sizes

domain. However, in this case AGCBM1 still performs significantly better than Random Deletion and Footprint-based methods except for the Auto domain.

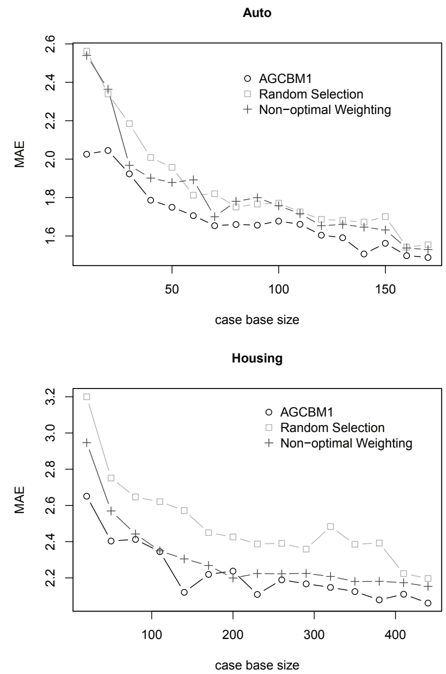

Figure 3: MAE of AGCBM1 and two variants in two sample domains

Method	Domain							
	Auto		MPG		Housing		Hardware	
	Min	Max	Min	Max	Min	Max	Min	Max
Random	p<.01	**p<.31**	p<.01	p<.02	p<.01	p<.01	p<.01	p<.01
FootPrint	p<.01	**p<.4**	p<.01	p<.03	p<.01	p<.01	p<.01	p<.01
Ablated1	p<.01	**p<.23**	p<.01	**p<.2**	p<.01	p<.01	p<.02	**p<.5**
Ablated2	p<.01	**p<.25**	**p<.4**	**p<.15**	p<.03	p<.03	p<.01	**p<.2**

Table 1: Significance for AGCBM1 outperforming alternatives

Conclusion and Future Work

This paper introduced AGCBM, an approach to condensing the case base size for domains in which adaptation knowledge is generated from cases, in which retention decisions reflect both cases' usefulness as source cases and their usefulness for generating adaptation rules on demand. An experimental evaluation of AGCBM1, a specification of the general method, showed improved accuracy, often by substantial margins, especially for high compression. An ablation study showed the benefit of AGCBM1's method of ranking cases to order their presentation to the CNN algorithm.

Future directions for extending this work include examining methods to increase efficiency and exploring how AGCBM could be applied to domains with symbolic features as input features or the target value, *e.g.*, for classification. Other directions include exploring different ways of combining case and adaptation competence values and comparing AGCBM's dynamic approach to adaptation rule generation to eager adaptation rule generation and retention methods applied in advance of system processing.

References

Aha, D.; Kibler, D.; and Albert, M. 1991. Instance-based learning algorithms. *Machine Learning* 6(1):37–66.

Angiulli, F. 2005. Fast condensed nearest neighbor rule. In *Proceedings of the twenty-second international conference on Machine learning*, 25–32. New York: ACM.

Brighton, H., and Mellish, C. 2001. Identifying competence-critical instances for instance-based learners. In *Instance Selection and Construction for Data Mining*, volume 608 of *The Springer International Series in Engineering and Computer Science*. Berlin: Springer. 77–94.

Brodley, C. 1993. Addressing the selective superiority problem: Automatic algorithm/model class selection. In *Proceedings of the Tenth International Conference on Machine Learning*, 17–24. Morgan Kaufmann.

Craw, S.; Massie, S.; and Wiratunga, N. 2007. Informed case base maintenance: A complexity profiling approach. In *Proceedings of the Twenty-Second National Conference on Artificial Intelligence*, 1618–1621. AAAI Press.

Delany, S., and Cunningham, P. 2004. An analysis of case-base editing in a spam filtering system. In *Advances in Case-Based Reasoning*. Berlin: Springer. 128–141.

Frank, A., and Asuncion, A. 2010. UCI machine learning repository. `http://archive.ics.uci.edu/ml`.

Hanney, K., and Keane, M. 1996. Learning adaptation rules from a case-base. In *Proceedings of the Third European Workshop on Case-Based Reasoning*, 179–192. Berlin: Springer Verlag.

Hanney, K. 1997. Learning adaptation rules from cases. Master's thesis, Trinity College, Dublin.

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

Jalali, V., and Leake, D. 2013a. A context-aware approach to selecting adaptations for case-based reasoning. In *Modeling and Using Context*. Berlin: Springer. 101–114.

Jalali, V., and Leake, D. 2013b. Extending case adaptation with automatically-generated ensembles of adaptation rules. In *Case-Based Reasoning Research and Development, ICCBR 2013*, 188–202. Berlin: Springer.

Leake, D., and Wilson, D. 2000. Remembering why to remember: Performance-guided case-base maintenance. In *Proceedings of the Fifth European Workshop on Case-Based Reasoning*, 161–172. Berlin: Springer.

Leake, D.; Smyth, B.; Wilson, D.; and Yang, Q., eds. 2001. *Maintaining Case-Based Reasoning Systems*. Blackwell. Special issue of *Computational Intelligence*, 17(2), 2001.

Lieber, J. 1995. A criterion of comparison between two case bases. In *Advances in Case-Based Reasoning*, volume 984 of *Lecture Notes in Computer Science*. Berlin: Springer. 87–100.

Lopez de Mantaras, R.; McSherry, D.; Bridge, D.; Leake, D.; Smyth, B.; Craw, S.; Faltings, B.; Maher, M.; Cox, M.; Forbus, K.; Keane, M.; Aamodt, A.; and Watson, I. 2005. Retrieval, reuse, revision, and retention in CBR. *Knowledge Engineering Review* 20(3).

Markovitch, S., and Scott, P. 1993. Information filtering: Selection mechanisms in learning systems. *Machine Learning* 10(2):113–151.

Racine, K., and Yang, Q. 1997. Maintaining unstructured case bases. In *Case-Based Reasoning Research and Development, ICCBR 1997*, 553–564. Berlin: Springer.

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.

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. San Mateo: Morgan Kaufmann.

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. 1998. Modelling the competence of case-bases. In *Proceedings of the Fourth European Workshop on Case-Based Reasoning*, 208–220. Berlin: Springer.

Smyth, B., and McKenna, E. 1999. Building compact competent case-bases. In *Proceedings of the Third International Conference on Case-Based Reasoning*, 329–342. Berlin: Springer Verlag.

Smyth, B. 1998. Case-base maintenance. In *Tasks and Methods in Applied Intelligence*, 507–516. Castellon, Spain: Springer.

Wilson, D., and Martinez, T. 2000. Reduction techniques for instance-based learning algorithms. *Machine Learning* 38(3):257–286.

Zhang, J. 1992. Selecting typical instances in instance-based learning. In *Proceedings of the Ninth International Workshop on Machine Learning*, 470–479. San Francisco: Morgan Kaufmann.