Extending Case Adaptation with Automatically-Generated Ensembles of Adaptation Rules

Vahid Jalali and David Leake

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

Abstract. Case-based regression often relies on simple case adaptation methods. This paper investigates new approaches to enriching the adaptation capabilities of case-based regression systems, based on the use of ensembles of adaptation rules generated from the case base. The paper explores both local and global methods for generating adaptation rules from the case base, and presents methods for ranking the generated rules and combining the resulting ensemble of adaptation rules to generate new solutions. It tests these methods in five standard domains, evaluating their performance compared to four baseline methods, standard k-NN, linear regression, locally weighted linear regression, and an ensemble of k-NN predictors with different feature subsets. The results demonstrate that the proposed method generally outperforms the baselines and that the accuracy of adaptation based on locally-generated rules is highly competitive with that of global rule-generation methods with much greater computational cost.

1 Introduction

Case-based reasoning (CBR) (e.g., Mantaras *et al.*, 2005) solves new problems by retrieving stored prior cases solving similar problems, and adapting their solutions to fit new circumstances, based on the differences between the new problem and problems addressed by the retrieved case(s). When CBR is applied to synthesis tasks in knowledge-rich domains, an important component of its success is the use of sophisticated case adaptation strategies. However, when CBR approach is applied to regression tasks, reliance on simple case adaptation is common. For example, k-Nearest Neighbor (k-NN) regression approaches often compute target values as a distance-weighted average of the values of the k cases closest to the input problem. Using simple adaptation helps to alleviate the knowledge acquisition problem for case adaptation knowledge for these tasks, and in practice can achieve good performance (e.g., [2]). However, the contrast between extensive focus on case adaptation in other CBR areas and the limited attention to richer adaptation methods for case-based regression raises the question of whether case-based regression performance could be improved by generating richer combination/adaptation rules automatically.

This paper presents new approaches for automatically augmenting adaptation capabilities for case-based regression, using only knowledge contained in the case base. Its

S.J. Delany and S. Ontañón (Eds.): ICCBR 2013, LNAI 7969, pp. 188–202, 2013.

primary contributions are methods for generating adaptation rules from local or global sets of cases, methods for applying ensembles of adaptation rules, and an experimental comparison of alternative strategies for using local and global information in both adaptation rule learning and rule application, which illuminates the relative performance benefits of local and global approaches.

The paper is organized as follows. Section 2 introduces the strategies we consider for generating adaptation rules and selecting the base cases from which the estimations are built. Section 3 introduces our approach, Ensemble of Adaptations for Regression (EAR), a general technique for augmenting k-NN for regression tasks by automatically generating adaptation rules, choosing which of many potentially applicable rules to apply, and using the resulting ensemble of rules for generating new solutions. It also describes the basic parameters of the approach, which adjust its use of local versus global information in selecting cases to adapt and generating adaptation rules from existing cases. Section 4 presents results of an evaluation comparing alternative versions of EAR with k-NN, linear regression, and locally weighted linear regression for estimating solutions in five sample domains. The study shows encouraging results for accuracy and for the ability to rely on local information, compared to more computationally expensive use of extensive global information, which suggests the practicality of lazy learning of adaptation rules based on local information. Section 5 compares related work on using ensemble techniques in CBR and knowledge-light methods for generating and applying adaptations for case-based regression tasks. Section 6 presents conclusions and future work.

2 Learning and Applying Ensembles of Adaptation Rules

Our basic approach to adaptation rule generation builds on the case difference heuristic approach proposed by Hanney and Keane [3] and further explored by others (e.g., [4,5]). The case difference approach builds new adaptation rules from pairs of cases and compares their problem parts (respectively, solution parts), and identifies their differences to generate a candidate rule mapping the observed difference in problems to the observed difference in solutions. For example, for predicting apartment rental prices, if two apartments differ in that one has an additional bedroom, and its price is higher, an adaptation rule could be generated to increase estimated rent when adapting a prior case to predict the price of an apartment with an additional bedroom. Applying the case difference approach depends on addressing questions such as which pairs of cases will be used to generate adaptation rules, how rules will be generated, and how the resulting rule set will be applied to new problems. In the next section, we discuss EAR's strategies for addressing these, and in Section 5 we compare these approaches to previous work.

EAR is a lazy approach to adaptation rule generation. Given an input problem, it generates ensembles of adaptations as needed, based on preselected criteria for (1) selecting a neighborhood of cases in the case base from which to generate solutions by adaptations, and (2) generating rules for adapting each of those cases, ranking the rules for each case and combining the values of the top r rules, and finally, combining the values generated for each of the cases to adapt. This process is summarized in Algorithm 1.

Algorithm 1. EAR's basic algorithm

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

$CasesToAdapt \leftarrow$ NeighborhoodSelection(Q,n,CB)
$NewRules$: $\leftarrow$ RuleGenerationStrategy(Q,$CasesToAdapt$,CB)
for c in $CasesToAdapt$ **do**
  RankedRules $\leftarrow$ RankRules($NewRules$,c,Q)
  $ValEstimate(c) \leftarrow$ CombineAdaptations($RankedRules$, c, r)
end for
return CombineVals($\cup_{c \in CasesToAdapt} ValEstimate(c)$)

2.1 Selecting Source Cases to Adapt

We consider three general alternatives for selecting the cases to adapt, defined by whether they use highly local, local, or global cases:

1. Nearest: Select only the single nearest neighbor to the query (1-NN)
2. Local: Select the k nearest neighbors to the query (k-NN, for a small value of k greater than 1)
3. Global: Select all cases in the case base

As we discuss in Section 5, adaptation learning methods using *nearest* and *local* case sets have been considered previously in CBR, but the global approach is seldom used. Because the global approach may consider cases quite dissimilar from the input query, its feasibility depends on the quality of the adaptation and combination strategies used.

2.2 Selecting Cases from which to Generate Adaptation Rules

For each case selected to be used as a source case for adaptation, we consider three options for selecting pairs of cases to be used to generate adaptation rules, as listed below. The strategies are described by their names, which have the form StartingCasesEndingCases, where StartingCases describes a set of cases for which rules will be generated, and EndingCases describes the cases to which each of the StartingCases will be compared. Each comparison results in a different rule, so a single starting case may participate in the formation of many rules.

1. Local cases–Local neighbors: Generating adaptation rules by comparing each pair of cases in the local neighborhood of the query.
2. Global cases–Local neighbors: Generating adaptation rules by comparing each case in the case base with its k nearest neighbors
3. Global cases–Global neighbors: Generating adaptation rules by comparing all cases in the case base

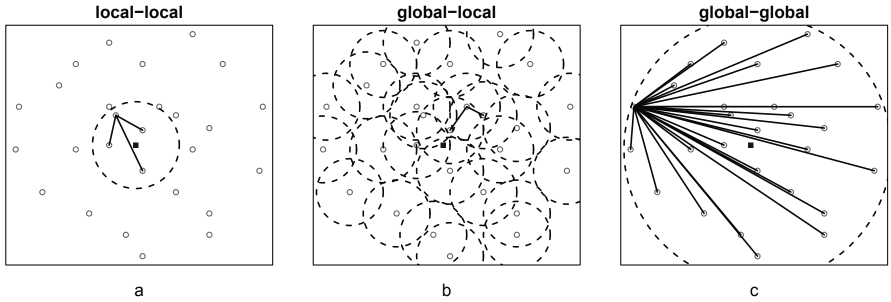

Fig. 1. Illustration of (a) Local cases–Local neighbors, (b) Global cases–Local neighbors, and (c) Global cases–Global neighbors

Figure 1 illustrates the three methods. In each figure, the input problem is at the center. Circles enclosed by dotted lines show neighborhoods of cases from which adaptations will be generated, and a sample point is connected to the cases with which it will be compared to generate adaptation rules.

Potential Tradeoffs. Combining each of the three selection strategies with one of the three adaptation generation strategies gives nine possible approaches. Each approach has potential ramifications for efficiency and accuracy of adaptation.

Ramifications for efficiency: The different methods provide different levels of efficiency. Using Global cases to generate adaptations from Local neighbors requires considering more rules than generating adaptations for local cases only. Using Global cases–Global neighbors, determining each adaptation requires $O(n^2)$ processing for a case base with n cases, which may be infeasible for large case bases.

A related ramification is the potential for lazy learning of adaptation rules as needed. The Global cases–Global neighbors approach requires processing all potential adaptations. If this is applied to the system's initial cases, a rule set can be stored for future use, avoiding re-calculation but potentially requiring considerable storage and—if kept static—not reflecting new cases added to the case base. Local cases–Local neighbors is amenable to a lazy approach with just-in-time generation of adaptation rules, which could enable incremental adaptation rule generation, with adaptation rule generation taking into account any new cases added to the case base in the region of the query.

Ramifications for accuracy: It is more difficult to anticipate the accuracy effects of the strategies. For example, one might hypothesize that generating adaptation rules from local cases would be beneficial because the adaptations are being generated from the same area of the domain space as the input query, making them more likely to properly address the differences between the input query and the base case(s). On the other hand, limiting the scope of adaptations to the context of the input query might sacrifice the benefit of considering distant cases corresponding to relevant adaptations. This raises the interesting question of locality of adaptation knowledge. Even if case characteristics for a particular case base are associated with particular regions of the case base, it is possible that the needed adaptation knowledge is still global: that the relationships between their feature changes and value changes may be similar regardless of region. This stance has long been taken implicitly in CBR systems which have been designed with a single set of adaptation rules applied in all parts of the case base. To our knowledge, the question of locality of adaptation knowledge has not been studied previously, and the following experiments shed some light on this question as well.

3 Using Ensembles of Adaptations

The methods described in the previous sections may result in the generation of many adaptation rules, especially for global–global rule generation. EAR's adaptation rule ensembles are composed of a subset of the selected rules, to increase adaptation efficiency. To select rules, EAR ranks them by the similarity of the current adaptation context and the context in which the rule was generated.

3.1 Defining Adaptation Context

After generating adaptation rules for an input query, EAR attempts to determine which of the generated rules are most relevant. It does this by considering both the similarity of the new query and the case for which the adaptation was generated, and the local adaptation characteristics of the case base, which we call the *adaptation context*. When selecting adaptations to apply to generate a solution for the query, EAR favors adaptations which have been generated for target problems in similar adaptation contexts. When global knowledge is used for generating the adaptations, for example, the cases used to generate an adaptation rule may be quite different from the query, but if the adaptation addressed similar differences, it may still be relevant.

Given a case C in the case base, EAR calculates its adaptation context as a vector based on comparing C to the N cases in a neighborhood containing its nearest neighbor cases. For each case feature, the covariance between the feature and the case solution is calculated over the set of cases in the neighborhood.

Let C_i^j and $Sol(C_i)$ denote the value of the j^{th} feature and the value of the i^{th} case respectively, $CaseMeanVal$ denote the mean of the values of the cases in the neighborhood, and $FeatureMeanVal_j$ represent the mean value of the j^{th} feature of the cases in the neighborhood. Then the j^{th} element of the covariance vector for case C is calculated as follows:

$$Cov_j(C) \equiv \frac{1}{N} \times \sum_{i=1}^{N} (C_i^j - FeatureMeanVal_j)(Sol(C_i) - CaseMeanVal) \quad (1)$$

If f represents the number of features, for any case C, we define $AdaptContext(C)$ to be the vector $(Cov_1, Cov_2, ...Cov_f)$.

3.2 Ranking Adaptation Rules

EAR's adaptation rule ranking considers two factors. The first is the similarity of the pair *query - source case to adapt* and the pair *target case - source case* from which the adaptation rule was generated. The second is the similarity of the adaptation context of the query to the adaptation context of the target case from which the adaptation rule was generated. However, if the adaptations are generated from local cases-local neighbors the second factor is discarded. The first factor favors adaptations generated to adapt similar pairs of cases. For each feature, EAR calculates the per-feature difference, based on a domain similarity metric, and generates a difference vector of those values.

The second factor reflects similarity of the adaptation context (as defined above), and compares the adaptation context vectors of the query and the target case. The rationale is that the same feature difference between two cases may require different adaptations in different parts of the case space, so favoring rules from similar adaptation contexts may improve adaptation results.

EAR's ranking method balances feature differences against adaptation context differences by taking the Hadamard (element-wise) product of the feature difference vector and the adaptation context vector. The ranking score is computed as the Euclidean distance between: (1) the Hadamard product of the adaptation context vector of the case to adapt and the difference vector for the case to adapt and the input query, and (2) the Hadamard product of the context vector of the adaptation rule and the vector representing feature differences of the composing cases of that rule.

More formally, suppose query Q is to be solved by adapting the case C_i. let Δ_i represent the difference vector of the features of the query Q and C_i, and let R_r be the problem part of the r^{th} adaptation rule. Let $\circ$ represent the Hadamard product of two vectors. Then the second component of EAR's rule scoring method is calculated as:

$$d(\Delta_i, R_r) \equiv distance((AdaptContext(C_i) \circ \Delta_i), (AdaptContext(C_r) \circ R_r)) \quad (2)$$

If $D(\Delta_i, R_r)$ is the distance between Δ_i and R_r, then the final ranking of adaptation rules is achieved by using a weighted average of D and d as:

$$RuleScore(R_r) \equiv a \times D(\Delta_i, R_r) + (1 - a) \times d(\Delta_i, R_r) \quad (3)$$

where $0 \leq a \leq 1$. The value of a is set to tune the ranking for different domains.

3.3 Estimating the Target Value from a Rule Ensemble

In its simplest form, k-NN estimates the value of a query by averaging the value of its k nearest neighbors. If Q is the query and $Est(Q)$ represents its estimated target value (solution), and $Sol(C_i)$ represents the known solution value of the i^{th} nearest neighbor of Q, then k-NN estimates the value of Q as:

$$Est(Q) \equiv \frac{\sum_{i=1}^{k} Sol(C_i)}{k} \quad (4)$$

For each base case C to be adapted to provide a value for a query, EAR computes a weighted average of the values proposed by each of the n highest-ranked adaptation rules generated for that case by Eq. 3. If $r_i, 1 \leq i \leq n$ are the n top-ranked adaptation rules in order of descending rank score,

$$SuggestedVal(C) = \sum_{i=1,n} \frac{Solution(r_i)}{i} \quad (5)$$

The value for the query is then simply

$$Solution(Q) \equiv \frac{\sum_{i=1}^{k} SuggestedVal(C_i)}{k} \quad (6)$$

4 Experimental Results

We conducted experiments to address the following questions about extending case adaptation with ensembles of automatically-generated adaptation rules:

1. Can using the automatically-generated ensembles of adaptations improve accuracy over using a single adaptation?
2. How does accuracy compare for adaptations generated from local vs. global knowledge?
3. How does EAR's accuracy compare to that of the baseline regression methods locally weighted linear regression and k-NN?
4. How does EAR's accuracy compare to that of case-based regression using standard feature subset ensemble methods?
5. How does EAR's rule process ranking (based on adaptation context and case similarity) affect its performance compared to the baselines of (1) random selection of adaptation rules and (2) considering case similarity only?

4.1 Data Sets and Experimental Design

Our experiments use five data sets from the UCI repository [6]:Automobile (A), Auto MPG (AM), Housing (H), Abalone (AB), Computer Hardware (CH). For all data sets, records with unknown values were removed. To enable comparison with linear regression, only numerical features were used. (Note that if the use of adaptation context in EAR is disabled, it could be used for symbolic features as well; including those potentially would have increased accuracy for EAR when local cases-local neighbors strategy is used for generating the adaptations). For each feature, values were standardized by subtracting that feature's mean value from each individual feature value and dividing the result by the standard deviation of that feature. Table 1 summarizes the characteristics of the test domains.

The experiments estimate the target value for an input query. For the Auto, MPG, Housing, Abalone and Hardware, the respective values to estimate are price, mpg,

Table 1. Characteristics of the test domains

Domain name	# features	# cases	Avg. cases/solution	sol. sd
Auto	13	195	1.1	8.1
MPG	7	392	3.1	7.8
Housing	13	506	2.21	9.2
Abalone	7	1407	176	1.22
Hardware	6	209	1.8	160.83

MEDV (median value of owner-occupied homes in $1000's), rings (for the Abalone data set we only selected cases with rings ranging 1-8), and PRP (published relative performance) respectively. Linear regression and locally weighted linear regression tests used Weka's [7] simple linear regression and locally weighted learning classes respectively. Accuracy is measured in terms of the Mean Absolute Error (MAE) and leave-one-out testing is used for all domains unless explicitly mentioned otherwise.

Hill climbing was used to select the best neighborhood size for each domain based on the training data for calculating adaptation context, for setting the weighting factor a Eqn. 3, and for determining the number of adaptations to consider. The number of adaptations used for different variants of EAR depending on the training data ranges from one for EAR9 to at most 40 for EAR1, EAR2 and EAR3. In all experiments Euclidean distance is used as the distance function in equation 2. Note that the use of contextual information is disabled for versions of EAR that use the local-local strategy to generate adaptations (i.e. EAR1, EAR4 and EAR7).

4.2 Performance Comparison

To address experimental questions 1–3, we conducted tests to compare the results achieved by each of the 9 versions of EAR, k-NN, linear regression (LR) and locally weighted linear regression (LWLR) in the sample domains. Table 2 summarizes the results, which we discuss below. Best values are indicated in bold.

4.3 Discussion of Results

Accuracy from Ensembles vs. Single Adaptations: In the experiments, EAR4 (local, local-local), EAR5 (local, global-local), EAR6 (local, global-global) and EAR9 (global, global-global) usually yield the best results, suggesting the benefit of generating adaptations based on multiple cases and selecting adaptations from their results to combine. For most methods, the tuning process on the training data determined that generating the final value from an ensemble of the top-ranked adaptations gave the best results.

Table 2. MAE of EAR, k-NN, LWLR and LR for the sample domains

Method	**Domains**				
	Auto (A)	MPG (AM)	Housing (H)	Abalone (AB)	Hardware (CH)
EAR1: nearest, local-local	1.77	2.23	2.21	0.79	31.32
EAR2: nearest, global-local	1.66	2.22	2.2	0.82	31.04
EAR3: nearest, global-global	2.15	2.22	2.23	0.95	38.25
EAR4: local, local-local	1.38	1.90	2.04	0.60	28.74
EAR5: local, global-local	1.44	**1.71**	**1.90**	0.60	28.8
EAR6: local, global-global	**1.36**	1.74	2.04	0.60	28.76
EAR7: global, local-local	4.95	4.99	4.22	0.93	78.06
EAR8: global, global-local	4.30	3.73	4.46	0.91	63.98
EAR9: global, global-global	1.37	1.95	2.25	**0.59**	**28.18**
k-NN	1.61	2.00	2.74	0.61	29.12
Locally Weighted LR (LWLR)	1.61	2.02	2.54	0.68	30.82
Linear Regression (LR)	2.62	2.55	4.53	0.62	51.91

For example, EAR4 (local, local-local) yields its best results (in all domains except Abalone) when usually five to nine adaptations are combined. There were some exceptions to the general pattern in favor of using ensembles of adaptations. For EAR9 (global, global-global) in most cases using one adaptation per case in the Auto, MPG and Housing domains yields best results (for the Hardware domain, often two cases are used). For the Abalone domain the optimal number of adaptations based on the training data is on the order of 20, but the difference between using one adaptation rule and greater numbers is minimal (1%).

Effect of Domain Characteristics on EAR's Performance: We observed that EAR showed less benefit for the Abalone data set than for the other data sets, with performance of EAR often comparable to k-NN. We hypothesize that the level of improvement from EAR over k-NN could be related to the diversity of case solutions in the case base.

If a relatively large number of cases share identical solutions in a domain, and the standard deviation of solutions is low, using an appropriate similarity measure in a retrieve-only system (e.g. k-NN) may be sufficient to generate good solutions with simple averaging combination, while with more diversity, more adaptation may be needed. Table 1, shows the average number of cases sharing the same solution and the standard deviation of the solutions in the sample domains, which shows that these characteristics of the Abalone data set are substantially different from the other data sets. However, more examination is needed.

Local vs. Global Knowledge for Generating Adaptations: Table 2 shows that in most domains, the performance of EAR4 (local, local-local) is competitive with the other versions of EAR, and is superior to the baseline methods, despite the fact that it uses limited information. For example, comparing EAR4 to the most global method, EAR9 (global, global-global), MAE's are 1.38 vs. 1.37, 1.9 vs. 1.95, 2.04 vs. 2.25, 0.60 vs. 0.59, and 28.74 vs. 28.18. Because it uses limited information, it is computationally much less expensive than the global methods. Thus the local method's performance at worst has a minimal accuracy penalty, and sometimes is substantially better. Also, it has the benefit of reducing computational cost and permitting a lazy approach to adaptation rule generation).

EAR7 (global, local-local) and EAR8 (global, global-local) usually yield the worst results. In those two methods all cases are considered as base cases for estimating the target value, so adaptation generated from neighbor cases may not be appropriate for addressing the differences between the input problem and the base cases.

EAR vs. LWLR and k-NN. In all domains, the performance of EAR4 surpasses or equals that of the baseline methods, sometimes substantially so. EAR4 has almost the same performance as k-NN in Abalone and Hardware domains. In all domains, one of the nine versions of EAR has the best performance.

In Auto, MPG and Housing domains that EAR4 shows higher accuracies compared to the other baseline methods, one side paired t-test is used to assess the significance of those results. The null hypothesis is always MAE of EAR4 being less than that of k-NN

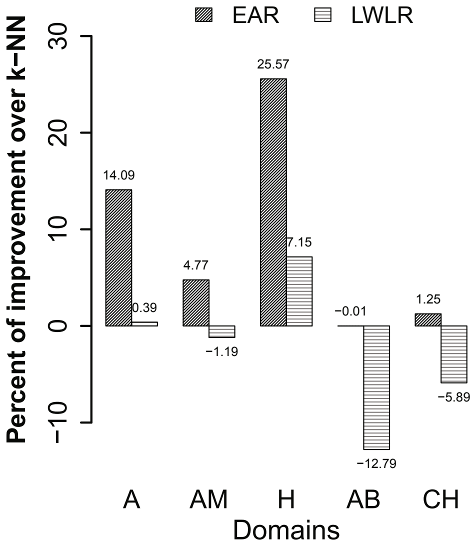

Fig. 2. Percent of improvement in MAE by EAR and LWLR over k-NN

and LWLR. For the comparison of EAR4 to k-NN in the Auto domain, $p<.007$, in MPG, $p<.062$ (so not significant), and in Housing, $p<.001$. Same values for comparing EAR4 versus LWLR are $p<.051$ (not significant), $p<0.05$ and $p<.001$ in the same order.

Figure 2 contrasts the relative improvement of EAR4 over k-NN (14%, 5%, 26%, 0% and 1%) with the relative improvement of LWLR over k-NN (0%, -1%, 7%, -13% and -6%) in the the Auto, MPG, Housing, Abalone and Hardware domains respectively.

EAR vs. Feature Subset Ensemble. As another baseline, we also compared EAR4's performance to a previously used approach for applying ensembles to CBR, feature subset ensembles (FSE). FSE uses a combination of k-NN predictors, each of which predicts based on a different subsets of case features (all subsets are of fixed size) [8]. The feature subsets are selected randomly with replacement (each subset includes at least two features), with each ensemble containing predictors based on 100 different subset of features, with evaluation by ten-fold cross validation. Both EAR4 and the feature subset ensemble methods were compared with their best parameter settings, as determined by hill climbing and leave-one-out testing on the training data for each fold. For feature subset ensembles, this determined the k value to use, and the number of features to use. For EAR4, this determined the number of base cases and adaptation rules to be used. For each domain, the local neighborhoods were set to contain the top 5% nearest neighbors of the input query. Learning was disabled for both methods. Table 3, shows Mean Absolute Error for k-NN, Feature Subset Ensemble (FSE) and EAR4 (local, local-local) on the test domains.

Table 3. MAE of EAR4, k-NN and the Feature Subset Ensemble method for the sample domains

Method	Domains				
	Auto (A)	MPG (AM)	Housing (H)	Abalone (AB)	Hardware (CH)
k-NN	1.62	2.06	2.67	**0.61**	30.3
FSE	1.51	2.28	2.48	0.7	27.51
EAR4	**1.42**	**1.84**	**2.01**	0.63	**25.79**

Table 4. MAE of EAR, k-NN, LWLR and LR for the sample domains

Method	Domains				
	Auto (A)	MPG (AM)	Housing (H)	Abalone (AB)	Hardware (CH)
EAR4: local, local-local	1.38	1.90	**2.04**	**0.60**	**28.74**
EAR6: local, global-global	**1.36**	**1.74**	**2.04**	**0.60**	28.74
Random: local, local-local	2.54	2.11	3.04	0.61	38.95
Random: local, global-global	3.87	2.43	3.29	0.61	72.86
distance only: local, global-global	1.55	1.86	2.11	0.61	30.68

The results in Table 3 show that EAR outperforms FSE in all test domains. For the Abalone domain, k-NN slightly outperforms both ensemble methods, which we hypothesize to be due to lack of domain diversity. Figure 3, shows the percent of improvement of EAR4 (local, local-local) and SFE over k-NN in the test domains.

4.4 Effect of Context-Based Rule Ranking

A final question is how much EAR's context-based adaptation rule ranking approach benefits performance. We tested this by an ablation study comparing EAR4 and EAR6's performance with three different ranking methods: (1) random ranking of adaptation rules, (2) rule ranking by case distance only, and (3) EAR's approach, balancing case similarity and adaptation context similarity.

As Table 4 shows, random ranking has the worst performance among other methods, with especially bad performance for the global-global methods, which generate more rules. The comparative difference appears to increase for domains with higher standard deviation (e.g. Hardware), and is lowest for Abalone, which has the largest average number of cases per unique solution and the lowest solution standard deviation. There the random method shows same performance as the distance only method.

Expanding the pool of adaptations with global methods decreases accuracy for distance-only method in nearly all domains, while EAR is more robust. This provides some support for the contextual information in EAR enabling it to select more appropriate adaptations from the global pool.

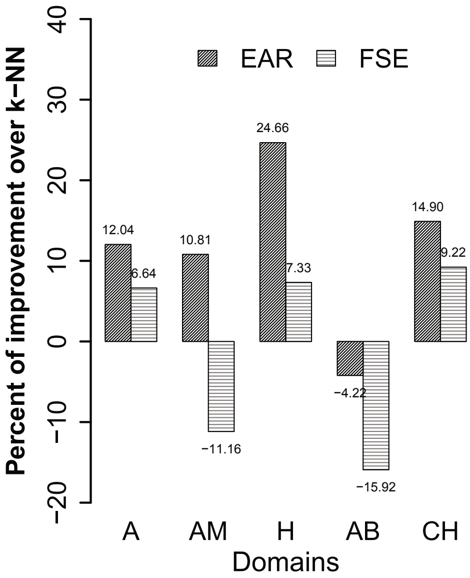

Fig. 3. Percent of improvement in MAE by EAR4 and LWLR over k-NN

5 Comparison to Previous Work

The EAR approach relates both to research on ensemble methods in CBR and on automatic adaptation rule generation for case-based regression.

5.1 Ensemble Methods in CBR

Ensemble methods aggregate results from a set of models. A number of general-purpose approaches have been proposed, such as Bagging [9], boosting [10] and random forests [11]. In CBR research, ensemble methods have been applied to improve accuracy by combining solutions from multiple subsets of a case base or from multiple case bases. For example, Cunningham and Zenobi [12] propose improving accuracy of nearest neighbor classifiers by using an ensemble of classifiers, each based on different feature subsets. Arshadi and Jurisica [13] present an ensemble method for combining predictions of a set of classifiers built based on disjoint subsets of cases from the original case base, for which the case features are selected locally by using logistic regression. Li and Sun [14] propose using an ensemble of CBR systems, with randomly generated feature subsets used for similarity assessment in each individual CBR system, and forming the final solution by combining the results of those individual systems. However, to our knowledge, previous CBR research has not considered the use of ensembles of case adaptation rules.

5.2 Learning Adaptations from the Case Base

Learning case adaptation knowledge is an active CBR research area, for which many approaches have been pursued. For reasons of space, we limit our discussion to methods which learn adaptations from cases in the case base for regression tasks, rather than more knowledge-intensive approaches for other types of domains.

Case Difference Heuristics. Wilke et al. [15] provide a starting point for knowledge-light approaches to learning adaptation knowledge by discussing different sources of knowledge in a CBR system and general issues for designing a learning algorithm. They use their framework for two different approaches of learning adaptation knowledge: weighted majority voting and case difference heuristic proposed by Hanney and Keane [3]. The latter approach investigated by Wilke et al. is similar to ours in that it generates adaptations based on case comparison. Though, their method uses different strategies for ranking rules (e.g. confidence rating for rules) and composing the final solutions compared to ours.

McSherry's [4] CREST (Case-based Reasoning for ESTimation) provides another approach to generating adaptations from case differences. Given a case to adapt, McSherry's difference heuristic attempts to retrieve a case which differs from the input query only in the value of a single feature, called the distinguishing attribute. Next, a pair of cases with the same values for the distinguishing attributes as the query and (respectively) the case to be adapted are retrieved, and the solution of the retrieved case is adjusted based on their difference. Because more than one similar case may be retrieved for an input query, the final estimation of the target value can be calculated by averaging different estimations, generated by the same method. McSherry's method is similar to EAR's local approach, in generating adaptations based on neighbors to the input query. However, CREST adjusts the solutions of each base case by applying a single adaptation, while EAR uses an ensemble of adaptations.

McDonnell and Cunningham [5] refine the case difference heuristic to address two problems. The first is that the effect of variations in feature values on the solution may differ according to the feature considered. The second is that the effect of variations in a feature value on the solution may depend upon the values of other case features. Their method generates adaptations by comparing the input query to nearby cases, selecting cases for which the gradient is similar to the target case (using local linear regression to approximate the gradients), and deriving adaptations from those cases. This approach is in the spirit of EAR's context-based approach but not applied to ensembles of adaptations.

Learning Adaptation Rules from Linear Regression. Patterson et al. [16] propose a rule acquisition process based on k-NN and regression analysis. Given a new problem, the k nearest neighbors are retrieved and combined in a new generalized case in which features are the distance-weighted average of the individual case features. The k nearest neighbors are also used to train a linear regression model for predicting the difference between case solutions, which is applied to the generalized case to predict the target value for the input. Like EAR, this method uses a lazy approach for generating adaptations; it differs in that it relies on linear regression and single adaptations for generating and applying adaptations, instead of case differences and ensemble of adaptations, respectively.

Other Adaptation Learning Models for Case-Based Regression. Adaptation learning for regression also includes methods not based on direct case comparisons. Policastro et al. [17] propose a method for learning and applying adaptation knowledge from a case base by using two components, estimators and combiner. As estimators they use a multi-layer neural network, an M5 regression tree, and a support vector machine. As combiners, they consider the same three techniques, applied to combine the estimators' values.

Craw et al. [18], Jarmulak et al. [19], and Wiratunga et al. [20] propose automated acquisition of adaptation knowledge by repeatedly partitioning the case base to form a small set of probe cases, retrieving k similar cases for each probe case, and building adaptation rules based on pairs of probe cases and their top k neighbors. For each set, their method creates rule sets, each one containing adaptation cases that concentrate on differences for a single feature. From those, their method selects rules whose decision tree indexes have above-average predictive accuracy. An initial solution is generated by averaging, with possible refinement by adaptation rules each addressing differences in a single feature.

6 Conclusions and Future Research

This paper has introduced EAR, an approach for automatically generating sets of adaptation rules from a case base based on case differences and selecting ensembles of adaptations to apply. An experimental evaluation of nine variants of the EAR approach showed that EAR variants generally increased accuracy over baseline case-based regression and linear regression approaches, and that rule generation based on local information was sufficient to obtain accuracy competitive with the best performance obtained. Likewise, an ablation study provided support for the benefit of EAR's context-based rule ranking approach.

Opportunities for future research include developing more sophisticated adaptation selection and combination techniques, exploring other ensemble methods for the generation and combination of adaptations, and examining how EAR could apply to knowledge-rich domains. Yet another direction for extending this work is considering the confidence of cases to adapt and the adaptation rules in EAR. That is to some extent explored in [21]. Also the question of comparative benefit of using local vs. global adaptations is an interesting one for future research.

References

1. Mantaras, R., McSherry, D., Bridge, D., Leake, D., Smyth, B., Craw, S., Faltings, B., Maher, M., Cox, M., Forbus, K., Keane, M., Aamodt, A., Watson, I.: Retrieval, reuse, revision, and retention in CBR. Knowledge Engineering Review 20(3) (2005)
2. Aha, D., Kibler, D., Albert, M.: Instance-based learning algorithms. Machine Learning 6(1), 37–66 (1991)

3. Hanney, K., Keane, M.: The adaptation knowledge bottleneck: How to ease it by learning from cases. In: Leake, D.B., Plaza, E. (eds.) ICCBR 1997. LNCS, vol. 1266, pp. 359–370. Springer, Heidelberg (1997)
4. McSherry, D.: An adaptation heuristic for case-based estimation. In: Smyth, B., Cunningham, P. (eds.) EWCBR 1998. LNCS (LNAI), vol. 1488, pp. 184–195. Springer, Heidelberg (1998)
5. 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 (LNAI), vol. 4106, pp. 91–105. Springer, Heidelberg (2006)
6. Frank, A., Asuncion, A.: UCI machine learning repository (2010), `http://archive.ics.uci.edu/ml`
7. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. SIGKDD Explor. Newsl. 11(1), 10–18 (2009)
8. Bay, S.D.: Combining nearest neighbor classifiers through multiple feature subsets. In: Proc. 15th International Conf. on Machine Learning, pp. 37–45 (1998)
9. Breiman, L.: Bagging predictors. Mach. Learn. 24(2), 123–140 (1996)
10. Schapire, R.E.: The strength of weak learnability. Mach. Learn. 5(2), 197–227 (1990)
11. Breiman, L.: Random forests. Mach. Learn. 45(1), 5–32 (2001)
12. Cunningham, P., Zenobi, G.: Case representation issues for case-based reasoning from ensemble research. In: Aha, D.W., Watson, I. (eds.) ICCBR 2001. LNCS (LNAI), vol. 2080, pp. 146–157. Springer, Heidelberg (2001)
13. Arshadi, N., Jurisica, I.: An ensemble of case-based classifiers for high-dimensional biological domains. In: Muñoz-Ávila, H., Ricci, F. (eds.) ICCBR 2005. LNCS (LNAI), vol. 3620, pp. 21–34. Springer, Heidelberg (2005)
14. Li, H., Sun, J.: Case-based reasoning ensemble and business application: A computational approach from multiple case representations driven by randomness. Expert Systems with Applications 39(3), 3298–3310 (2012)
15. Wilke, W., Vollrath, I., Althoff, K.D., Bergmann, R.: A framework for learning adaptation knowledge based on knowledge light approaches. In: Proceedings of the Fifth German Workshop on Case-Based Reasoning, pp. 235–242 (1997)
16. Patterson, D., Rooney, N., Galushka, M.: A regression based adaptation strategy for case-based reasoning. In: Proceedings of the Eighteenth Annual National Conference on Artificial Intelligence, pp. 87–92. AAAI Press (2002)
17. Policastro, C.A., Carvalho, A.C., Delbem, A.C.: A hybrid case adaptation approach for case-based reasoning. Applied Intelligence 28(2), 101–119 (2008)
18. Craw, S., Jarmulak, J., Rowe, R.: Learning and applying case-based adaptation knowledge. In: Aha, D.W., Watson, I. (eds.) ICCBR 2001. LNCS (LNAI), vol. 2080, pp. 131–145. Springer, Heidelberg (2001)
19. Jarmulak, J., Craw, S., Rowe, R.: Using case-base data to learn adaptation knowledge for design. In: Proceedings of the 17th International Joint Conference on Artificial Intelligence, IJCAI 2001, vol. 2, pp. 1011–1016. Morgan Kaufmann Publishers Inc. (2001)
20. Wiratunga, N., Craw, S., Rowe, R.: Learning to adapt for case-based design. In: Craw, S., Preece, A.D. (eds.) ECCBR 2002. LNCS (LNAI), vol. 2416, pp. 421–435. Springer, Heidelberg (2002)
21. Jalali, V., Leake, D.: On deriving adaptation rule confidence from the rule generation process. In: Delany, S.J., Ontañón, S. (eds.) ICCBR 2013. LNCS, vol. 7969, pp. 179–187. Springer, Heidelberg (2013)