A Context-Aware Approach to Selecting Adaptations for Case-Based Reasoning

Vahid Jalali and David Leake

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

Abstract. Case-based reasoning solves new problems by retrieving cases of similar previously-solved problems and adapting their solutions to fit new circumstances. The case adaptation step is often done by applying context-independent adaptation rules. A substantial body of research has studied generating these rules automatically from comparisons of prior pairs of cases. This paper presents a method for increasing the context-awareness of case adaptation using these rules, by exploiting contextual information about the prior problems from which the rules were generated to predict their applicability to the context of the new problem, in order to select the most relevant rules. The paper tests the approach for the task of case-based prediction of numerical values (case-based regression). It evaluates performance on standard machine learning data sets to assess the method's performance benefits, and also tests it on synthetic domains to study how performance is affected by different problem space characteristics. The results show the proposed method for context-awareness brings significant gains in solution accuracy.

1 Introduction

Problem solving by case-based reasoning (CBR) retrieves cases capturing the solutions of similar past problems and adapts their solutions to fit new circumstances [1]. How to generate knowledge to guide the case adaptation process is a classic challenge for case-based reasoning. Often, case-based reasoning systems adapt cases based on a limited set of context-independent rules hand coded by domain experts [2]. Given the cost and difficulty of generating case adaptation knowledge, the CBR community has investigated using more knowledge-light approaches to acquire case adaptation rules by machine learning, and especially learning by comparing cases in collection of prior cases (the "case base") and inferring how differences in problem descriptions suggest solution differences—which in turn show how solutions should be adapted to address the differences between new problems and retrieved cases [3,4]. Such methods are highly promising. However, they can result in a large set of possible rules for adapting any particular difference, which raises the question of how to select the adaptation rules to apply. Traditionally, selection of adaptation rules has been based only on the feature differences between old and new situations, with little attention to

P. Brézillon, P. Blackburn, and R. Dapoigny (Eds.): CONTEXT 2013, LNAI 8175, pp. 101–114, 2013.

the context in which the differences appear. This paper presents a new method for making case adaptation more context-aware, by favoring rules which were generated for not only similar differences, but for differences which arose in similar contexts.

This paper provides both a general perspective on the role of context in case adaptation for CBR and a specific method for context-aware adaptation rule selection when CBR is applied to the task of numeric regression, for which the goal is to estimate a numeric value associated with a set of input parameters. When case-based reasoning is applied to the regression task, the set of inputs is considered the "problem" to solve, and the "solution"—the output value—is estimated by retrieving similar past problems from the case base and building a solution based on the solution values of those similar cases. The prior solution values are "adapted" according to the differences between the problems they solved and the new problem. For example, a real-world application of case-based regression is real estate appraisal [5], for which the task is to predict the value of a property, based on the values of similar properties. If the most similar prior case is a smaller house, a new house's price should be predicted by adjusting the prior house's price to reflect the size difference.

Obviously, when rules are generated automatically from case comparison, many overlapping and inconsistent rules might be generated. Consequently, how to select the adaptation rules to apply to a particular problem becomes an important question. This paper presents a case study of a new context-based approach to selecting adaptation rules for case-based regression, for adaptation rules generated automatically based on case differences. It tests performance of the approach compared to five alternative methods, on six standard machine learning data sets and on synthetically generated domains designed to study how particular domain characteristics affect performance of the approach.

The paper is organized as follows. Section 2 presents an overview of case-based regression and previous work on using knowledge-light approaches to generate case adaptation rules for case-based regression, focusing on a popular method for generating adaptation rules, the *case difference heuristic* approach [3]. Section 3 explains our method for context-aware application of case adaptation rules, which has been implemented in the system CAAR (Context-Aware Adaptation Retrieval). Section 4 discusses the motivations for the design of the synthetic data sets, provides details about the synthetic and standard data set characteristics and reports the results of empirical evaluation of the candidate methods on a set of synthetic and real world data sets analyzing and comparing the performance of the methods under different circumstances. Section 5 presents conclusions and future research directions.

2 Applying Case-Based Reasoning to Regression Tasks

2.1 Overview of Case-Based Regression

Case-based regression computes the solution value of a new problem based on the values of k "nearest neighbor" cases (for some predefined integer k) retrieved

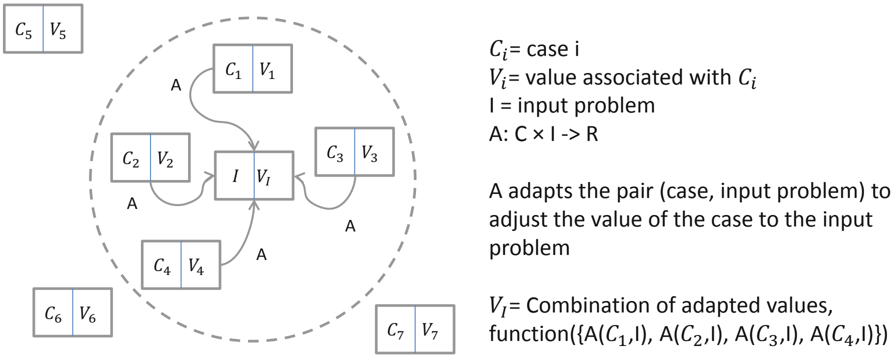

Fig. 1. Illustration of the generic case-based regression process

from the case base. Given an input problem description (generally in the form of a vector of feature values), the nearest neighbor cases are those whose problem descriptions are most similar to the input problem, according to a predefined similarity metric. To calculate the solution value, the values of the nearest neighbor cases may be adapted, based on the differences between the problems they addressed and the new problem. The values are then combined by a combination function (e.g., into a weighted average in which the contributions of each case are weighted by the similarity of their problem to the input problem). Figure 1 illustrates 4-*NN*, with A designating the adaptation function.

2.2 Generating Adaptation Rules by the Case Difference Heuristic

Given the potential difficulty and cost of generating case adaptation rules by hand, it is desirable to generate them automatically. A highly influential approach to automatically generating case adaptation rules for case-based regression is the *case difference heuristic* method, introduced by Hanney and Keane [3]. This approach generates adaptation rules from prior cases, by comparing pairs of cases in the case base. For each pair, the approach compares the problem specifications of the two cases, generating a description of their differences which we refer to as "case difference vector". Often, this vector simply records the numerical differences between the case features. This vector is used as the applicability condition for the new rule; the new rule will be applied when a new input problem and a retrieved case have similar differences in each of their features.

For each pair, the approach also compares the solutions, generating a description of their solution differences. The observed difference becomes the adaptation part of the new rule; the rule adjusts the value of the prior case by this difference when the rule applies. For example, for real estate price prediction, if two

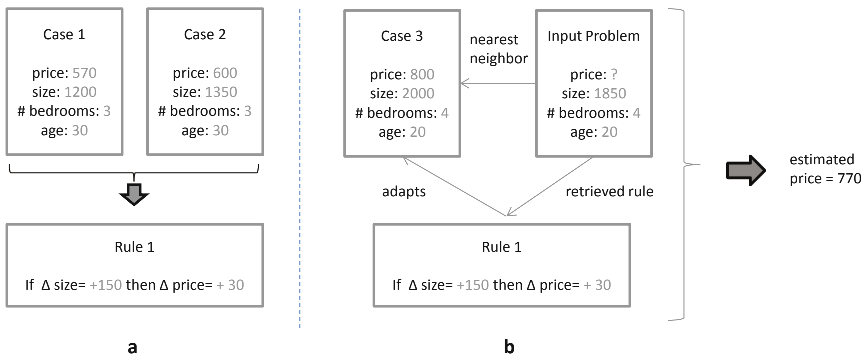

Fig. 2. illustration of problem differences and solution differences

apartments' descriptions differ only in that one is 150 square feet larger than the other, and the larger apartment's rent is $30 more per month, this suggests the rule that a 150 square foot size increase should increase the rent by $30. Part a of Fig. 2 depicts the generation of this rule (rule 1) from two cases, case 1 and case 2; part b depicts the application of rule 1 to a new case, case 3.

We note that the previous example rule is extremely simplified, and that many alternative rules might apply. For example, the adjustments might depend on percent changes, or correspond to a more complicated formula; how to address these issues is beyond the scope of this paper but is addressed elsewhere in the literature (e.g., [6]). Other case adaptation approaches for regression include alternative work on case difference heuristics [7], using linear regression for adapting the solutions [8], using a committee of machine learning methods [9].

2.3 Characterizing Context for the Case Adaptation

The importance of context is becoming widely recognized in artificial intelligence, but how to precisely define and characterize context in particular areas remains challenging. Dey [10] proposes that context is "any information that can be used to characterize the situation of an entity;" Brézillon [11] defines it as "what constrains a problem solving without intervening in it explicitly" and observes that context provides guidance for focusing attention in different tasks and that sometimes the "contextual reasoning is local reasoning."

In case-based reasoning research, adaptation rules have generally reflected only case differences, not the context in which those differences were observed. However, in some domains, the needed adaptations may vary substantially with context (e.g., in the real estate domain, the effect of the size of a lot on price may vary strongly based on whether the property being sold is in a city or a rural area, so adaptations should be sensitive to the location of the property).

Table 1. Comparison of Related Approaches

Method	Context Representation	Limited set of base cases	Focus of context
McDonnell [12]	Gradient Vector	False	Input query
Jalali [13]	Covariance Vector	True	Case to adapt
CAAR	Gradient Vector	True	Input query and case to adapt

Some previous research on case-based reasoning for regression has attempted to consider context in case adaptation. McDonnell and Cunningham [12] define the context of a point in problem space by approximating the rate of change (i.e., the gradient) of the regression system's target value function at that point. We note that their approach only considers the context for the input problem and for the corresponding case used to generate the adaptation rule.

Our own previous work [13] introduced EAR (Ensembles of Adaptations for Regression), in which the context of adaptation problems is characterized in terms of covariance vectors for the case to adapt and the corresponding case used to generate an adaptation rule. EAR selects adaptation rules to apply by doing a pair-wise multiplication of each component of two vectors. The first is the context vector which represents the covariance between the input features and the cases' values. The second is the case difference vector which represents the differences in the problem specification of the pair of input problem and case to adapt and pair of generating cases of the rule to apply. The distance between the vector calculated for the pair (input problem, case to adapt), and for the pair of the two cases used to generate the adaptation rule, is considered to measure their contextual similarities.

In addition to the contextual similarities, the similarities between the input problem and previous problems are calculated as the distance between the case difference vectors of the rule to apply (i.e. the case difference vector of the pair of cases used to generate the rule) and the pair of input problem and the case to adapt. The final rank score for an adaptation rule is generated by combining its case-based and contextual similarities to the pair of input problem-case to adapt by using a weighted average.

This paper presents a characterization of context for selecting learned adaptation rules that considers both the local situation of the problem to be adapted and the local situation of the cases from which the rules were generated, considering the changes that the gradient predicts in the case solutions, based on their feature differences. Table 1 summarizes the major differences between the two previous methods and the one proposed in this paper, implemented in CAAR. In addition, CAAR includes methods for refining adaptation rule retrieval and generation methods to make both more local, as well as for reducing computational cost by fixing the cases to adapt and focusing on the context-aware retrieval of adaptations for the selected set of cases to adapt.

3 CAAR

We hypothesize that performance of case adaptation can be improved by refining the treatment of adaptation context in two ways:

1. *Maximizing locality of data used in rule generation:* By restricting the cases used to generate adaptation rules to nearby cases, this aims to draw both cases from the same context, so that the relationship between the cases will give rise to meaningful rules.
2. *Enriching the context description:* By using context information to characterize both the similarity of the input problem and case to adapt, and the similarity of the case pair to the case pair used to generate the adaptation rule, this aims to select more relevant cases to adapt and rules to apply.

CAAR's algorithm respects the first condition by

1. First fixing the cases to adapt, choosing them to be the top nearest neighbors of the input problem, and then
2. Generating the adaptation rules to apply to the cases to adapt on demand, by comparing each case to adapt with its top nearest neighbors, and favoring rules addressing similar contexts.

The main focus of this paper is the second point, enriching the context description, which is described below.

3.1 CAAR's Adaptation Selection

CAAR selects adaptations to apply by ranking the candidate adaptations based on the similarity of the current adaptation context to the adaptation context in which the rule was generated, as follows. Let Q represent the input problem and C_b a case whose solution must be adapted to provide a solution to Q. Let C_i and C_j be the composing cases of the adaptation rule $R_{i,j}$ and $R_{j,i}$, where $R_{i,j}$ is a candidate for adjusting the value of case C_b to provide a value for Q. CAAR ranks candidate adaptations based on the similarity of two contexts: The context of the input problem and the corresponding composing case of an adaptation rule with regard to their differences with the case to adapt and its corresponding composing case of the adaptation and the context of the case to adapt and the corresponding composing case of the adaptation rule with regard to same changes respectively.

The ranking score is calculated by the function $score : rules \times cases \times problems \rightarrow R^+$, calculated by:

$$score((C_i, C_j, C_b, Q)) = contextSim((C_i, C_j, C_b, Q)) + contextSim((C_j, C_i, Q, C_b)) \quad (1)$$

As input, *score* takes the two cases used to generate the adaptation rule being assessed, the case to be adapted, and the input problem. It calls the function *contextSim* twice, once to determine the appropriateness of the adaptation rule $R_{i,j}$ to adapt C_b to the query Q (based on the similarity of the context in which the rule was generated to the adaptation context defined by the relationship between C_b and Q), and once to assess context-based appropriateness of the reverse rule ($R_{j,i}$), applied to adapt Q to C_b. By considering both directions, the computation takes into account both the context at the query (via the first term) and at the case to be adapted (via the second term). The final score is the sum of both terms.

The function *contextSim* is defined as follows. Like *score*, *contextSim* takes four arguments, the two cases used to generate the adaptation rule $R_{i,j}$, a case to adapt, and a query. Let $\nabla(C)$ represent the gradient vector around the case C, $\mathit{Diff}((C_i, C_j))$ represent the feature differences of the ordered pair of cases C_i and C_j, $\cdot$ be the dot product, and K be a function for tuning the range of results. The *contextSim* function is calculated as:

$$contextSim((C_i, C_j, C_k, C_l)) = K(\mid Diff(C_i, C_j)cdot\nabla(C_i) - Diff(C_k, C_l) \cdot \nabla(C_k) \mid) \tag{2}$$

For example, if it is desired that the ranking score of Eq. 1 generate a higher score given one very high and one very low underlying similarity than given two medium level underlying similarities, K could be set to an exponential function, to scale the raw values such that extremal values have more weight.

3.2 Applying the Selected Adaptation Rules

Let Q represent the input problem and R_i represent the i^{th} adaptation rule in the ranked list generated using Eq. 1. Then CAAR's case adaptation adjusts the value of the case to adapt, C_b, by the average of the solution changes proposed by the top r adaptations, as follows:

$$adjustedVal(C_b, Q) = \sum_{i=1,r} \frac{1}{r} \times proposedAdjustment(R_i) \tag{3}$$

For k the number of selected cases to adapt to generate the solution, we use the algorithm we introduced in [13] to estimate the final solution, as follows:

$$finalEstimate(C_b, Q) = \sum_{i=1,k} \frac{1}{k} \times adjustedVal(C_{b,i}, Q) \tag{4}$$

Algorithm 1 summarizes the entire process.

4 Evaluation

Our evaluation addressed four questions:

1. How does the accuracy of CAAR compare to that of the baseline methods locally weighted linear regression, k-NN, and EAR?

Algorithm 1. Case-based regression with context-aware adaptation retrieval's basic algorithm [13]

Input:
Q: input problem
k: number of base cases to adapt to solve query
r: number of rules to be applied per base case
CB: case base
R: set of existing adaptations
Output: Estimated solution value for Q

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

2. How does CAAR's consideration of context at both the input case and the case to adapt affect performance, compared to considering context only at one or the other?
3. How is the accuracy of the candidate methods affected by increasing the density of case base coverage of the problem space? (Density will normally be correlated to case base size.)
4. How do changes in domain regularity (i.e., the lack of value fluctuations associate with different contexts) affect the accuracies of the candidate methods?

We expect that either increasing case base size or increased regularity will improve performance of all methods, because increased case base size increases the likelihood of finding cases to adapt from regions with similar characteristics. On the other hand, we expect increasing the rate of fluctuations in the context to make it harder for all methods to generate accurate estimations. However, we expect this to affect locally weighted learning more drastically than CAAR, especially for sparser case bases: We predict that when there is a shift in the changes of the target function (e.g. descending and then ascending), taking the average of the training data will be more accurate than fitting a locally learned linear model. Therefore, we expect to see an increase in the accuracy of CAAR compared to that of locally weighted linear regression for higher frequencies.

4.1 Data Sets

We tested CAAR's method on both synthetic and real world data sets. Synthetic data sets were used to enable precise control over the data characteristics, for addressing questions 3 and 4. Real world data sets were used to assess performance of CAAR's method compared to other candidate methods under more realistic scenarios in domains with more features.

Standard data sets: The standard data sets included four from the UCI repository [14]:Automobile (A), Auto MPG (AM), Housing (H), Computer Hardware (HW) and two from Luis Torgo's Regression data sets [15]: Stock (S) and CPU. For all data sets, records with unknown values were removed. To enable comparison with linear regression, only numeric features were used in the experiments. 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.

Synthetic data sets: The synthetic data sets were generated by a sinusoidal model. This model was chosen for two reasons: First, because its behavior in different regions corresponds to different contexts (given our treatment of context in terms of gradient and the changes in the gradient of the sine function over the X axis), and second, because it provides a repetitive pattern of context changes, so that rules generated from different parts of the domain space can still have similar contexts. Cases in the synthetic datasets all have a single input feature, which during data generation is associated to the value given by $sin(\frac{f}{2\pi}x)$, where f is a frequency value held constant for a given data set. Case input feature values are in the range [0,100], selected randomly with a uniform distribution. Data sets were generated for all combinations of 20 case base sizes (from 50 to 525 cases, step size 25) and 10 frequencies (from 0.021 to 0.083, which gave rise to sine waves covering from approximately 2-8 complete periods as x varied from 0 to 100). This gave rise to a total of 200 synthetic data sets.

4.2 Experimental Design

The experiments estimate the target value for an input query. In all cases Mean Absolute Error is used for assessing accuracy. Leave-one-out testing and ten fold cross validation are used for conducting the experiments on the synthetic and real world data sets respectively. Candidate methods tested for generating estimations are k-NN, locally weighted linear regression (LWLR), EAR and CAAR.

For the Auto, MPG, Housing, Hardware, Stock and CPU data sets the respective values to estimate are price (the reported values are the actual prices divided by 1000), mpg, MEDV (median value of owner-occupied homes in $1000's), PRP (published relative performance), the company stock price and portion of time that cpu runs in user mode respectively. For the synthetic data sets, the value to predict is assigned to the cases based on their feature value, as explained in section 4.1.

The k-NN procedure and locally weighted linear regression were implemented using WEKA's [16] IBk and locally weighted learning (using the linear regression class as the base learner) classes. EAR is the method "EAR4" introduced in [13].

For each method and data set, parameters for each regression method were tuned using hill climbing and leave-one-out testing on the training data. The tuned parameter for the k-NN is k, the number of cases to consider; the tuned parameter for LWLR methods is the number of neighbor cases for building the estimation. For EAR and CAAR, tuning set the number of cases to adapt for

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

Method	Domains					
	Auto (A)	MPG (AM)	Housing (H)	Hardware (HW)	Stock (S)	CPU
k-NN	1.6	2.1	2.72	31.5	0.47	2.1
LWLR	1.64	1.87	2.22	26.4	0.51	1.9
EAR	1.43	1.93	2.14	25.64	**0.43**	1.93
CAAR1	1.44	1.78	2.01	26.4	0.53	1.98
CAAR2	1.58	1.82	1.98	28.2	0.54	2
CAAR	**1.35**	**1.77**	**1.91**	**25.24**	**0.43**	**1.87**

each problem and the number of adaptations to apply. When k-NN and LWLR were tuned, there was no limit on the number of cases to be used for building the estimations and models. The number of base cases for EAR was limited to the minimum of ten or top 2.5 % cases in the case base and the maximum number of adaptations to be applied per case is respectively limited to the number of adaptation rules generated from those base cases (following the rationale of [13], omitted here for reasons of space). The number of base cases for CAAR is also limited to the minimum of ten and the top 2.5% cases in the case base and the number of applied adaptations per base case is limited to 150. The scaling function K in Eq. 1 was set to the identity function.

4.3 Experimental Results

Standard Data Sets: Experiments on standard data sets were used to address evaluation question 1, how the accuracy of CAAR compares to that of the baseline methods locally weighted linear regression, k-NN, and EAR, and question 2, how the consideration of context of both input query and case to adapt affects performance, versus only considering context at one or the other, as in previous work. Table 2 lists the mean absolute error of the methods for the six methods and six data sets. CAAR1 and CAAR2 are ablated versions of CAAR, respectively considering only the context of the input problem or only the context of the case to adapt.

CAAR has the highest accuracy in all data sets, and outperforms its ablated versions, demonstrating the value of CAAR's more extensive consideration of context. k-NN has the lowest accuracy in four of the six domains. For four of the six data sets EAR outperforms locally weighted linear regression.

Figure 3 shows the percent of improvement in MAE for CAAR, EAR and LWLR over k-NN. Improvement of CAAR over k-NN ranges from 9% to 30%. Using a one side paired t-test with 95% confidence interval, and null hypothesis that the MAE of LWLR is less than that of CAAR, in the Auto domain $p<.001$, in the MPG domain $p<.038$, in the Housing domain $p<.001$, in the Hardware domain $p<.3$ (not significant), in the Stock domain $p<.001$ and in the CPU domain $p<.001$.

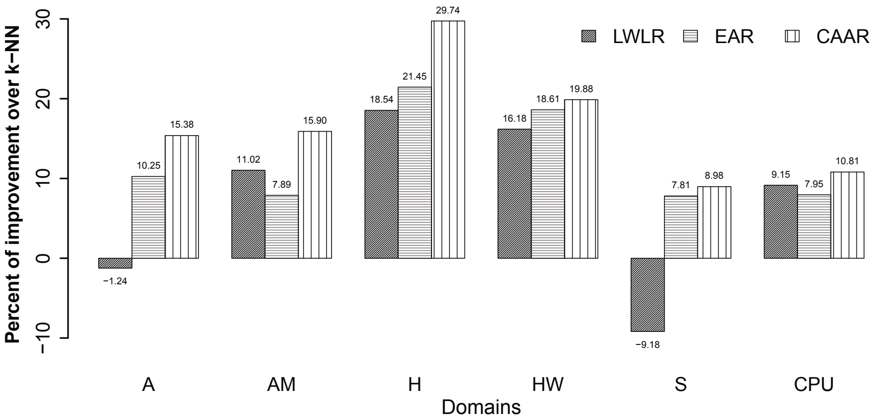

Fig. 3. Percent improvement in MAE of CAAR, EAR and LWLR over k-NN for the real world data sets

Synthetic Data Sets: Tests on synthetic data sets were used to explore Question 3, how the accuracy of the candidate methods is affected by increasing the density of case base coverage of the problem space, and Question 4, how changes in domain regularity (i.e., the level of fluctuations across different contexts) affect the accuracies of the candidate methods. Figure 4 shows the MAE of CAAR's estimates for the synthetic domains as a function of the frequency of domain changes and case base size. To show the whole spectrum of MAEs, a logarithmic scale is used. Figure 4 shows that increasing case density decreases MAE, and increasing frequency increases MAE. The explanation is that increased case base coverage increases the likelihood of CAAR being able to select prior cases within a similar context, and that higher frequencies decrease the size of regions with similar context, increasing likelihood of generating new adaptation rules from cases in different contexts.

Fig. 5 provides some representative examples from tests on the synthetic data. Part a of Fig. 5 fixes a representative synthetic data set frequency (0.049) and shows how the number of cases in the case base affects relative performance at that frequency of EAR, LWLR and CAAR compared to k-NN (lines have been added between points for visibility only). Increasing case-base size increases accuracy of all methods compared to k-NN, but CAAR always shows the best performance followed by LWLR and EAR.

Part b of Fig. 5 fixes case base size at a representative size, 150 cases, and illustrates performance as a function of frequency. Increasing frequency causes the relative advantage of EAR, LWLR and CAAR over k-NN to decrease,

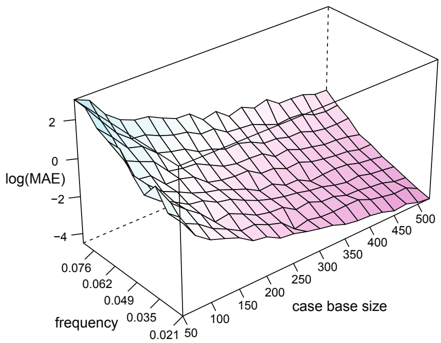

Fig. 4. MAE of CAAR on the synthetic data sets

Table 3. MAEs of k-NN, LWLR, EAR and CAAR methods for different synthetic data sets with 150 cases

Method	frequency									
	0.0208	0.0278	0.0347	0.0417	0.0486	0.0556	0.0625	0.0694	0.0764	0.0833
k-NN	1.87	2.43	2.41	2.53	3.42	3.45	3.87	4.61	6.27	7.91
LWLR	0.18	0.2	0.22	0.36	0.43	0.59	0.62	1.25	2.21	3.59
EAR	0.70	0.78	1.00	1.10	1.54	1.57	2.12	2.77	4.60	5.74
CAAR	**0.16**	**0.17**	**0.16**	**0.33**	**0.33**	**0.47**	**0.48**	**0.86**	**1.56**	**2.36**

but the loss for CAAR is less than for the other two methods. Table 3 shows the actual mean absolute errors for these results.

Part c of Fig. 5 shows the percent of improvement of CAAR compared to LWLR for frequency 0.049. CAAR shows an improvement ranging from 7% to 35%, for different case base sizes. However, there is no clear pattern. Part d of Fig. 5 shows relative improvement of CAAR compared to LWLR for a a case base of 150 cases. Here increasing the frequency increases the relative benefit of CAAR, with up to 34% improvement over LWLR when the frequency is maximum. We hypothesize that this is because higher frequencies result in higher fluctuations in the values of cases in local neighborhoods, which can make the locally fitted linear model inaccurate, but CAAR's reuse of the differences derived from similar contexts in the case base mitigates this problem to a certain degree.

Parts b and d show that on the synthetic data, CAAR's use of regularities with previous problems enables it to make more accurate estimations compared to LWLR, which supports its approach for regression tasks in domains with fairly regular patterns of past problem-solution pairs. Using a one-side paired t-test with 95% confidence interval, and null hypothesis that the MAE of LWLR and k-NN is less than that of CAAR, in all synthetic domains $p<.001$.

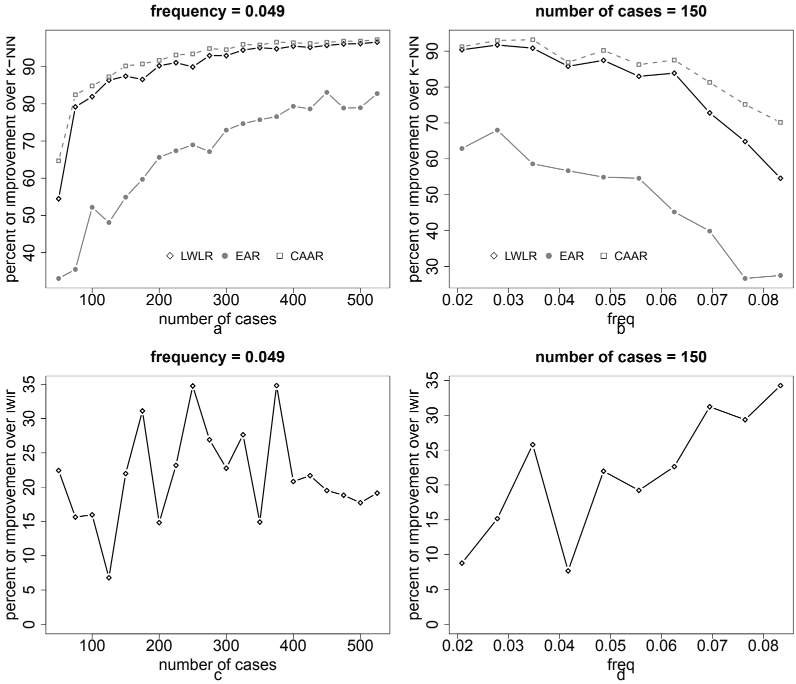

Fig. 5. comparison of the candidate methods performance on synthetic data sets

5 Conclusion and Future Research

This paper has introduced a method for using contextual information to improve the accuracy of case-based regression. The approach considers two types of context, the context of the input problem, and the context in which candidate case adaptation rules were generated, and uses these types of context to select cases to adapt to solve problems and to select automatically-generated adaptation rules to adapt those cases. Context is based on the gradient of the locally weighted fitted linear model at each point of the domain space.

An experimental evaluation of the new method compared to four baseline methods and two ablations, in 200 synthetic and six real-world domains showed that the approach can improve the estimation accuracies, and that considering both problem context and adaptation context is more beneficial than considering either alone.

Future work includes exploration of whether also considering the level of confidence in particular solutions can be used to improve context calculations (cf. [17]). Long term goals include extending this general approach to apply to domains with symbolic features, as well as to develop methods for defining and using adaptation context in tasks such as classification, and eventually for more knowledge-rich tasks such as case-based planning.

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. Leake, D.: Learning adaptation strategies by introspective reasoning about memory search. In: Proceedings of the AAAI 1993 Workshop on Case-Based Reasoning, pp. 57–63. AAAI Press, Menlo Park (1993)
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. 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)
5. Bonissone, P., Cheetham, W.: Financial applications of fuzzy case-based reasoning to residential property valuation. In: Proceedings of the Sixth IEEE International Conference on Fuzzy Systems, vol. 1, pp. 37–44 (1997)
6. Hanney, K.: Learning adaptation rules from cases. Master's thesis, Trinity College, Dublin (1997)
7. 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)
8. 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)
9. 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)
10. Dey, A.: Understanding and using context. Personal Ubiquitous Computing 5(1), 4–7 (2001)
11. Brézillon, P.: Context in problem solving: A survey. The Knowledge Engineering Review 14(1), 1–34 (1999)
12. 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)
13. 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)
14. Frank, A., Asuncion, A.: UCI machine learning repository (2010), `http://archive.ics.uci.edu/ml`
15. Torgo, L.: Lus torgo - regression data sets, `http://www.dcc.fc.up.pt/~ltorgo/Regression/DataSets.html`
16. 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)
17. 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)