Supporting Case-Based Reasoning with Neural Networks: An Illustration for Case Adaptation

David Leake, Xiaomeng Ye and David Crandall

Luddy School of Informatics, Computing, and Engineering, Indiana University, Bloomington, IN 47408, USA

Abstract

Case-based reasoning (CBR) is a knowledge-based reasoning and learning methodology that applies prior cases—records of prior instances or experiences—by adapting their lessons to solve new problems. The CBR process enables explainable reasoning from few examples, with minimal learning cost. However, the success of CBR depends on having appropriate similarity and adaptation knowledge, which may be hard to acquire. This paper illustrates the opportunity to leverage neural network methods to reduce the knowledge engineering burden for case-based reasoning. It presents an experimental example from ongoing work on refining the case difference heuristic approach to learning case adaptation knowledge by applying neural network learning.

Keywords

case adaptation, case-based reasoning, knowledge acquisition, neural networks, hybrid systems

1. Introduction

Case-based reasoning (CBR) is a methodology for reasoning and learning in which agents reason by retrieving and adapting the lessons of prior episodes [1, 2, 3, 4, 5]. A major inspiration for CBR models came from observations of human reasoning [2, 6]. Human experts—and others—are reminded of past experiences as they encounter new problems. The sharing of "war stories" is a common way experts transmit knowledge. Motivations for applying CBR include easing knowledge acquisition, both because cases may be easier to elicit than rules [3] and because, in some domains, cases are captured routinely as a byproduct of other processes, providing a readily-available knowledge resource [7]. CBR also provides multiple choices for where to place domain knowledge, enabling knowledge engineers to focus knowledge capture effort wherever most convenient. CBR models have been developed for many knowledge-rich tasks and have been widely applied [8, 9, 10, 11].

However, even when case acquisition and engineering are straightforward, case-based reasoning requires additional knowledge sources that may be difficult to acquire. Most notably, the knowledge used to adapt prior solutions to new circumstances is often captured in rule-based form and may be hard to generate. For many years, acquiring case adaptation knowledge has been seen as a key challenge for case-based reasoning [3, 12]. The difficulty of acquiring case

In A. Martin, K. Hinkelmann, H.-G. Fill, A. Gerber, D. Lenat, R. Stolle, F. van Harmelen (Eds.), Proceedings of the AAAI 2021 Spring Symposium on Combining Machine Learning and Knowledge Engineering (AAAI-MAKE 2021) - Stanford University, Palo Alto, California, USA, March 22-24, 2021.

0000-0002-8666-3416 (D. Leake); 0000-0002-2289-1022 (X. Ye)

adaptation knowledge has led to numerous CBR applications that focus primarily on retrieval [13, 14], functioning as extended memories for a human rather than as full problem-solvers.

The difficulty of capturing adaptation knowledge has led to interest in how case adaptation knowledge can be learned. The most widely used approach, called the case difference heuristic [12], generates rules by comparing pairs of cases, ascribing differences in their solutions to differences in the problems they address. The method generates new rules that adjust solutions analogously when a retrieved case differs from a new problem in a similar way. This approach has proven useful, but has depended on human expertise to define problem characterizations and to determine how to generalize the observed differences.

This paper argues for increasing the flexibility and accuracy of case difference heuristic models by using a neural network to learn how to process a given difference. Following on seminal work by Liao, Liu and Chao [15], it implements a case difference heuristic model by using a neural network to learn how to process a given difference. However, rather than relying only on the difference, as in their work, our approach also provides the neural model with the problem context in which the adaptation is performed. We present experiments illustrating its benefits over both CBR baselines and a neural net baseline. Because an important benefit of adaptation is extending the ability of a CBR system to address novel queries, we specifically test performance for such queries. The results support the benefit of the approach in that setting.

The paper first highlights the complementary strengths of case-based reasoning and neural network methods, which make it appealing to achieve benefits from both by using network methods to support case-based reasoning. It then sketches the steps of the case-based reasoning process, the sources of knowledge on which it depends, and the case difference heuristic approach to learning case adaptation knowledge. It next presents a preliminary case study on exploiting a neural network to determine solution differences for the case difference heuristic. Finally, it considers broader opportunities for synergies between case-based reasoning and network methods.

2. Complimentary Strengths of Case-Based Reasoning and Network Methods

The contrasting properties of CBR and network models make it appealing to use network methods to support CBR. CBR is appealing because it can function successfully with very limited data, and because the ability to place knowledge in multiple "knowledge containers" (as described below) can facilitate development of knowledge-rich systems. In addition, it is a lazy learning method with inexpensive learning: CBR systems learn by simply storing new cases, without generalization until (and only if) needed to process a new problem.

Neural network models, in contrast, do not easily exploit prior knowledge. They depend on large data sets and are an eager learning method, generalizing at training time, making learning expensive. However, they offer the ability to achieve high performance in a knowledge light way. Thus they are promising for learning from data to support CBR.

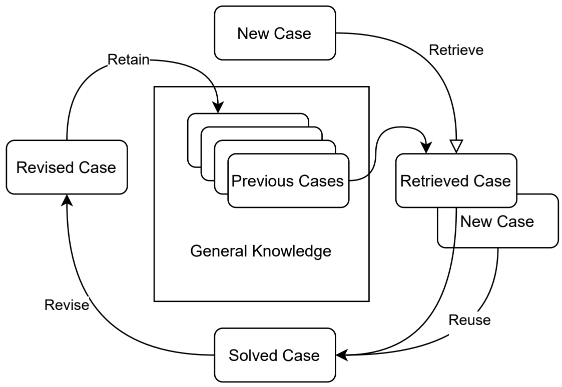

Figure 1: The CBR cycle. Image based on Aamodt and Plaza [1]

3. Case-Based Reasoning and Knowledge

3.1. The CBR Cycle

The CBR process is a cycle in which problems are presented to the system for processing steps often described as *retrieve, reuse, revise*, and *retain*. The most relevant prior case is retrieved, its solution is reused—matched to the new situation—and then revised—adapted to fit—and finally, retained—stored as a new case, learned by the system. The process is illustrated in Figure 1.

The case-based reasoning process uses multiple forms of knowledge, commonly referred to as the CBR knowledge containers [16]: representational vocabulary, case knowledge, similarity knowledge, and case adaptation knowledge. The knowledge containers can be seen as overlapping, in the sense that placing knowledge in one can decrease the need for knowledge in another. For example, increasing the case base size can decrease the need for adaptation knowledge, if the added cases enable retrieving cases more similar to incoming problems (which reduces the need for adaptation). The ability to choose where to place knowledge provides flexibility for knowledge acquisition from humans and by automated learning methods.

3.2. Acquiring Case Adaptation Knowledge

Acquiring case adaptation knowledge is a classic hard problem for case-based reasoning. Case adaptation knowledge is often encoded in the form of rules, whose effectiveness may depend on quality of a domain theory. Early case-based reasoning research invested extensive effort to develop case adaptation knowledge (e.g., [17]). The difficulty of generating case adaptation knowledge was a serious impediment to the development of CBR systems with rich reasoning, and prompted development of case-based aiding systems which functioned as external memories, retrieving cases to present to the user without adaptation [13]. Later work recognized the potential of learning methods to capture case adaptation knowledge. These included the generation of rules by decision tree learning [18, 19], and the use of case-based reasoning for the adaptation process itself [20, 21, 22, 23]

The most influential adaptation rule learning approach is the *case-difference heuristic* (CDH) approach. This knowledge-light approach generates adaptation knowledge using cases in the case base as data (e.g. [12, 24, 25, 26, 27, 28]). The case difference heuristic generates rules for adapting retrieved cases to fit new problems, using cases in the case base. Given a pair of cases, it calculates the difference between their problem descriptions (generally represented as feature vectors) and the difference between their solutions (generally numerical values for regression tasks). From the pair, a rule is generated. The rule encodes that when an input problem and retrieved case have a problem difference similar to the one from which the rule was generated, the solution for the retrieved case should be adjusted by the previously observed solution difference. For example, in the real estate price prediction domain, a rule might be generated from two similar apartments, one a two-bedroom and the other a three-bedroom, to adjust the price given an additional bedroom. Normally, human knowledge is used to determine how the adjustment should be done (e.g., a fixed or percent increment), and the process relies on the assumption that the old and new contexts will be sufficiently similar for the rule to apply to future cases.

Liao, Liu and Chao [15] have applied deep learning to learn differences to assign to the solution of a retrieved case for regression problems. Their method presents the problem difference of two cases to a network which has been trained on pairs to output solution differences. Craw et al. [29] showed that with careful tuning of feature weights, superior performance can be achieved by taking more context into account for the case difference heuristic. We are investigating the use of network methods to avoid the tuning step when adding context to the case-difference heuristic approach.

4. An Illustration from Case Adaptation

To investigate the effect of using a network to learn adaptations taking into account the context, we conducted an initial experiment. Liao, Liu and Chao [15] tested neural network adaptation for the NACA 0012 airfoil dataset [30] from the UCI repository [31]. Their results showed that neural networks can learn adaptations for a CDH approach in that domain. Our experiment compares five different systems: a k-NN system with $k = 1$, which can be seen as a CBR system with no case adaptation, a k-NN system with $k = 3$, which can be viewed as a CBR system with very simple adaptation (solution averaging), a CBR system using adaptation rules generated using the case difference heuristic ("normal CDH"), inspired by Craw et al. [29], a CBR system using a neural network to learn rules from CDH and carry out adaptation ("network CDH"), and, as a further baseline for comparison, a NN system that solves the regression problem directly.

The design of the network CDH system builds on the model of of Liao et al. [15], but differs in two respects. First, in addition to taking as input the problem differences, it takes as input the problem of the retrieved case, which provides context for the adaptation. Second, in addition to being trained on pairs of similar cases, it is trained on pairs of random cases, enabling generation of rules for larger differences (cf. Jalali and Leake [32]). Our experimental procedure differs from theirs in testing on data sets for which we restrict the available training cases so that the test query is always novel.

4.1. Implementations

The NN system is implemented as a feedforward neural network of four fully connected layers. Depending on the task domain, there is minor variation in the number of neurons per layer. The system is trained until the validation error converges.

The CBR system with normal CDH is implemented following Craw et al. [29]. A pair of cases is compared to produce an adaptation example, within which one of the two cases' problem descriptions is used as a context, indicating that a problem difference in such a context can lead to such a solution difference. This system is denoted as "**CBR + normal CDH**" and implemented as follows:

- Case retrieval: Training cases are stored in a case base. Given a query, the case retrieval process finds the most similar case from the case base using 1-NN.
- Case adaptation: During training, adaptation examples are assembled from pairs of training cases and stored in an adaptation case base CB_A. During testing, the problem difference between the query and the retrieved case is calculated. The problem description of the retrieved case is used as the context. Then a non-optimized 1-nearest neighbor algorithm retrieves the most similar adaptation example. This solution difference is added to the retrieved solution to produce the final solution.

The second system, the CBR system using CDH assisted by a neural network, denoted as "**CBR + network CDH**", and is based on Craw et al. [29] and Liao et al. [33]. Following the latter, a neural network NN_a is trained from cases treated as adaptation examples, as follows:

- Case retrieval: Training cases are stored in a case base. Given a query, the case retrieval process finds the most similar case from the case base using 1-NN. This is the same as in CBR + normal CDH.
- Case adaptation: During training, pairs of training cases are used to train an adaptation neural network NN_a to produce a solution difference given a problem difference and a context. During testing, the problem difference between the query and the retrieved case is calculated. The problem description of the retrieved case is used as the context. Then NN_a uses the problem difference and the context to propose a solution difference. This solution difference is added to the retrieved solution to produce the final solution.

For a given task domain, the required NN system might vary (e.g., more neurons might be needed if a case's problem description contains many features). No matter the variation of the NN system, the CBR + network CDH system always uses the same structure for NN_a.

We note that the experiments use a minimalistic design for all three systems. A CBR system can take many forms involving design choices such as retrieval, adaptation, case maintenance, user feedback and intervention, etc.; similarly, a NN system can vary by using different layers, numbers of neurons, activation functions, and connectivity. The CBR + network CDH and CBR + normal CDH systems are trained on the same adaptation examples, and the CBR + network CDH and NN systems use the same neural network structure. Our choices of models are based on the goal of a simple yet fair comparison, where all models are given the same case base and similar computational power.

All experiments are done under a constrained setting previously used by Leake and Ye [34], in which each test case is "forced" to be novel: the training phase is done **after** a test case is chosen, so the systems are only allowed to train on not-too-similar cases. More specifically:

- Before the systems are trained, a test case is chosen from the test set to be the query.
- The top ncr (standing for number of cases removed) neighbors of the query are identified and temporarily removed from the case base.
- The systems train using the trimmed case base:
  - The NN system is trained on 90% of the trimmed case base with the remainder used as the validation data set. The NN system is trained until its validation error converges.
  - The k-NN system is provided with the trimmed case base as training examples.
  - The CBR system uses the above k-NN with $k = 1$ as its case retrieval process. The CBR system trains its adaptation knowledge in a process inspired by Jalali and Leake [32]. From the trimmed case base, the CBR system assembles n pairs of a case and its nearest neighbor, using rp (standing for random pairs) pairs of randomly chosen cases.
- After the training phase, each system is tested on the query.

4.2. Experiment on Airfoil Data Set

For comparison with the results of Liao et al. [33], we performed the above experiment for the airfoil self-noise data set. In this data set, a problem description X is a vector of 5 numeric attributes describing wind and the airfoil blade section, and a solution description y is the sound level of the noise generated by the airfoil blade. The data set contains 1503 cases, 10% of which are used as the test cases. We use $rp = 5000$ and ncr is chosen from the range of $\{100, 200, 300, 400, 500\}$.

4.2.1. Experimental Results

The results are shown in Table 1. As ncr increases, all systems suffer to some extent because the queries become harder to solve. The system with the best result for each ncr is highlighted. We note that 3-NN consistently outperforms 1-NN, presumably because multiple retrievals decrease the influence of a potentially misleading nearest case. CBR + network CDH consistently outperforms 1-NN and 3-NN, which is expected because of the ability to do better adaptation. CBR + normal CDH performs poorly through all experiments. Given the better performance of CBR + network CDH, we hypothesize that the poor performance is due to inability to reliably select the right adaptation. A similar effect was observed by Craw et al. [29], where a suitable technique was needed to retrieve the best adaptation example. The NN system consistently

Table 1
Average MSE of systems for different values of ncr on the Airfoil dataset.

	Number of cases removed (ncr)				
	100	200	300	400	500
3-NN	1.083	1.229	1.387	1.600	1.742
1-NN	1.374	1.698	1.845	2.184	2.403
CBR + network CDH	0.484	0.693	0.824	1.016	**1.168**
NN	**0.409**	**0.549**	**0.749**	**0.864**	1.267
CBR + normal CDH	1.175	1.893	1.919	2.522	2.593

outperforms all other systems, and the CBR system ranks second, except when $ncr = 500$ and the CBR + network CDH ranks first.

In this data set, there are plenty of samples for values in each dimension, and many cases share the same attributes. In such a setting the NN system can learn to solve novel queries. When enough cases are removed to impair the NN system, the adaptation knowledge and overall performance of CBR + network CDH are also impaired.

4.3. Experiment on Car Features and MSRP Data Set

The next experiment is carried out on the Car Features and MSRP Data Set from Kaggle [35]. A problem description X contains fifteen numeric features such as engine horse power, and nominal features such as make and model. A solution description y is the price of a car. For cars sold in 2017, Manufacturer Suggested Retail Price (MSRP) is used. For older cars, True Market Value is collected from edmunds.com.

4.3.1. Experimental Settings

The original data set contains about 12000 cases. We cleaned the data by dropping rows with missing values. Nominal attributes were transformed into one-hot encodings. Additionally, we removed 4000 cases which share the same attributes with other cases but have slightly different solutions. We also removed extreme outlier cases (the rare cases with a solution price above 600,000). The cleaned data set contains 6629 cases, each with 1009 dimensions. The high dimensionality is due to the variety of values in nominal attributes, which are converted into one-hot encodings.

As in previous experiments, 10% of the cases are used as test queries. We use $rp = 10000$, and ncr is chosen from the range of $\{0, 1, 2, 10, 50, 100\}$. Differently from previous experiments, we evaluate systems when $ncr = 0$. Due to the time cost of our special testing procedure, we only test 50 random queries per experiment when $ncr \neq 0$.

4.3.2. Experimental Results

The test results are shown in Table 2. The best systems have comparable performance when $ncr = 0$ or $ncr = 1$. The CBR system substantially outperforms all other systems when $ncr \geq 2$.

Table 2
Average MSE of systems for different values of *ncr* on the Car Dataset.

	Number of cases removed (ncr)					
	0	1	2	10	50	100
3-NN	0.106	0.216	0.560	1.623	1.477	1.768
1-NN	0.065	0.040	0.497	1.677	1.527	2.039
CBR + network CDH	**0.029**	**0.030**	**0.049**	**0.257**	**0.237**	**0.256**
NN	0.035	0.080	0.108	0.413	0.544	0.560
CBR + normal CDH	0.076	0.067	0.489	1.672	1.487	1.973

Due to the high dimensionality, removing cases heavily impacts the quality of the nearest neighbor retrieval, as shown by the k-NN systems when $ncr \geq 2$. Without similar cases, the NN system cannot learn general knowledge about the query even if a minimal number of cases is removed, as shown by the NN system when $ncr \geq 2$. Nonetheless, we see the CBR system performs exceptionally well for novel queries in such a high dimensional data set. The general knowledge learned by the NN system may be less suitable to this novelty, while the adaptation knowledge learned by the CBR + network CDH system from $(n + rp)$ pairs of cases is less affected. Finally, we note that CBR + network CDH is essentially adapting the results of 1-NN. By comparing the two rows, we notice that often 1-NN performs poorly but the adaptation process often successfully estimates a correct result.

5. Opportunities for Using CBR Knowledge to Benefit Deep Networks

Additional opportunities for synergies between case-based reasoning and deep learning come in the reverse direction: how case-based reasoning may support deep learning.

Research on case-based reasoning supporting deep learning has primarily focused on using CBR systems and DL systems in parallel, to exploit the availability of similar cases when assessing network conclusions. Gates, Kisby, and Leake [36] propose pairing CBR and DL systems to assess solution confidence. Much CBR research has advanced the idea of "twin systems" that pair CBR and DL for explainability, as described in a survey by Keane and Kenny [37].

We see additional opportunity for fruitful pairings. One of the most knowledge-rich components of many CBR systems is the case adaptation knowledge component. This paper has discussed methods for decreasing the knowledge acquisition burden for case adaptation knowledge. Once adaptation knowledge has been acquired, either by human knowledge engineering or automatically, it becomes a resource that can be used in other contexts. We plan to explore the application of case adaptation knowledge to adapt the solutions generated by network methods.

6. Conclusion and Next Steps

Case-based reasoning provides benefits of explainability and the ability to reason effectively from small data sets, but suffers from the difficulty of obtaining knowledge to adapt cases. This paper has illustrated how a network approach can alleviate this knowledge acquisition problem, using an approach that augments prior work by considering the problem context in addition to the difference between cases. Experiments support improved performance, especially on novel queries, for which supporting adaptations with a neural network provides better performance than directly performing the task with the baseline neural network.

A next step will be to extend the CDH approach by exploiting the strength of deep learning to generate feature descriptions. Rather than relying on a network to learn the appropriate difference for a rule to apply, we intend first to use a deep network to derive the features to use to represent problems and solutions, and apply the case difference heuristic to learn adaptation rules for that new representation. This approach will use machine learning to refine both the vocabulary knowledge container and the adaptation knowledge container.

Acknowledgments

This material is based upon work supported in part by the Department of the Navy, Office of Naval Research under award number N00014-19-1-2655. We gratefully acknowledge the helpful discussions of the Indiana University Deep CBR team.

References

[1] A. Aamodt, E. Plaza, Case-based reasoning: Foundational issues, methodological variations, and system approaches, AI Communications 7 (1994) 39–52.

[2] J. Kolodner, Case-Based Reasoning, Morgan Kaufmann, San Mateo, CA, 1993.

[3] D. Leake, CBR in context: The present and future, in: D. Leake (Ed.), Case-Based Reasoning: Experiences, Lessons, and Future Directions, AAAI Press, Menlo Park, CA, 1996, pp. 3–30. Http://www.cs.indiana.edu/˜leake/papers/a-96-01.html.

[4] R. López de Mántaras, D. McSherry, D. Bridge, D. Leake, B. Smyth, S. Craw, B. Faltings, M. Maher, M. Cox, K. Forbus, M. Keane, A. Aamodt, I. Watson, Retrieval, reuse, revision, and retention in CBR, Knowledge Engineering Review 20 (2005).

[5] M. Richter, R. Weber, Case-Based Reasoning - A Textbook, Springer, 2013.

[6] D. Leake, Cognition as case-based reasoning, in: W. Bechtel, G. Graham (Eds.), A Companion to Cognitive Science, Blackwell, Oxford, 1998, pp. 465–476.

[7] W. Mark, E. Simoudis, D. Hinkle, Case-based reasoning: Expectations and results, in: D. Leake (Ed.), Case-Based Reasoning: Experiences, Lessons, and Future Directions, AAAI Press, Menlo Park, CA, 1996, pp. 269–294.

[8] W. Cheetham, I. Watson, Fielded applications of case-based reasoning, The Knowledge Engineering Review 20 (2005) 321–323.

[9] M. Göker, R. Howlett, J. Price, Case-based reasoning for diagnosis applications, The Knowledge Engineering Review 20 (2005) 277–281.

[10] A. Holt, I. Bichindaritz, R. Schmidt, P. Perner, Medical applications in case-based reasoning, Knowledge Eng. Review 20 (2005) 289–292.

[11] S. V. Shokouhi, P. Skalle, A. Aamodt, An overview of case-based reasoning applications in drilling engineering, Artificial Intelligence Review 41 (2014) 317–329.

[12] K. Hanney, M. Keane, Learning adaptation rules from a case-base, in: Proceedings of the Third European Workshop on Case-Based Reasoning, Springer, Berlin, 1996, pp. 179–192.

[13] J. Kolodner, Improving human decision making through case-based decision aiding, AI Magazine 12 (1991) 52–68.

[14] I. Watson, Applying knowledge management: Techniques for building corporate memories, Morgan Kaufmann, San Mateo, CA, 2003.

[15] C.-K. Liao, A. Liu, Y. Chao, A machine learning approach to case adaptation, 2018 IEEE First International Conference on Artificial Intelligence and Knowledge Engineering (AIKE) (2018) 106–109.

[16] M. Richter, Introduction, in: M. Lenz, B. Bartsch-Spörl, H.-D. Burkhard, S. Wess (Eds.), CBR Technology: From Foundations to Applications, Springer, Berlin, 1998, pp. 1–15.

[17] K. Hammond, Case-Based Planning: Viewing Planning as a Memory Task, Academic Press, San Diego, 1989.

[18] S. Craw, Introspective learning to build case-based reasoning (CBR) knowledge containers, in: P. Perner, A. Rosenfeld (Eds.), Machine Learning and Data Mining in Pattern Recognition, volume 2734 of *Lecture Notes in Computer Science*, Springer, 2003, pp. 1–6.

[19] S. Shiu, D. Yeung, C. Sun, X. Wang, Transferring case knowledge to adaptation knowledge: An approach for case-base maintenance, Computational Intelligence 17 (2001) 295–314.

[20] S. Craw, J. Jarmulak, R. Rowe, Learning and applying case-based adaptation knowledge, in: D. Aha, I. Watson (Eds.), Proceedings of the Fourth International Conference on Case-Based Reasoning, Springer Verlag, Berlin, 2001, pp. 131–145.

[21] D. Leake, A. Kinley, D. Wilson, Learning to integrate multiple knowledge sources for case-based reasoning, in: Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, Morgan Kaufmann, 1997, pp. 246–251.

[22] D. Leake, J. Powell, Mining large-scale knowledge sources for case adaptation knowledge, in: R. Weber, M. Richter (Eds.), Proceedings of the Seventh International Conference on Case-Based Reasoning, Springer Verlag, Berlin, 2007, pp. 209–223.

[23] M. Minor, R. Bergmann, S. Gorg, Case-based adaptation of workflows, Information Systems 40 (2014) 142–152.

[24] V. Jalali, D. Leake, Extending case adaptation with automatically-generated ensembles of adaptation rules, in: Case-Based Reasoning Research and Development, ICCBR 2013, Springer, Berlin, 2013, pp. 188–202.

[25] N. McDonnell, P. Cunningham, A knowledge-light approach to regression using case-based reasoning, in: Proceedings of the 8th European conference on Case-Based Reasoning, ECCBR'06, Springer, Berlin, 2006, pp. 91–105.

[26] D. McSherry, An adaptation heuristic for case-based estimation, in: Proceedings of the Fourth European Workshop on Advances in Case-Based Reasoning, EWCBR '98, Springer-Verlag, London, UK, UK, 1998, pp. 184–195.

[27] W. Wilke, I. Vollrath, K.-D. Althoff, R. Bergmann, A framework for learning adaptation knowledge based on knowledge light approaches, in: Proceedings of the Fifth German

Workshop on Case-Based Reasoning, 1997, pp. 235–242.
[28] M. D'Aquin, F. Badra, S. Lafrogne, J. Lieber, A. Napoli, L. Szathmary, Case base mining for adaptation knowledge acquisition, in: Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI-07), Morgan Kaufmann, San Mateo, 2007, pp. 750–755.
[29] S. Craw, N. Wiratunga, R. Rowe, Learning adaptation knowledge to improve case-based reasoning, Artificial Intelligence 170 (2006) 1175–1192.
[30] T. Brooks, D. Pope, M. Marcolini, Airfoil Self-noise and Prediction, NASA reference publication, National Aeronautics and Space Administration, Office of Management, Scientific and Technical Information Division, 1989.
[31] D. Dheeru, E. Karra Taniskidou, UCI machine learning repository, 2017. URL: http://archive.ics.uci.edu/ml.
[32] V. Jalali, D. Leake, Enhancing case-based regression with automatically-generated ensembles of adaptations, Journal of Intelligent Information Systems (2015) 1–22.
[33] C. Liao, A. Liu, Y. Chao, A machine learning approach to case adaptation, in: 2018 IEEE First International Conference on Artificial Intelligence and Knowledge Engineering (AIKE), 2018, pp. 106–109.
[34] D. Leake, X. Ye, On combining case adaptation rules, in: K. Bach, C. Marling (Eds.), Case-Based Reasoning Research and Development, Springer, 2019, pp. 204–218.
[35] Kaggle, Car features and MSRP, 2016. Data retrieved from Kaggle, https://www.kaggle.com/CooperUnion/cardataset.
[36] L. Gates, C. Kisby, D. Leake, CBR confidence as a basis for confidence in black box systems, in: Case-Based Reasoning Research and Development, ICCBR 2019, Springer, Berlin, 2019, pp. 95–109.
[37] M. Keane, E. Kenny, How case based reasoning explained neural networks: An XAI survey of post-hoc explanation-by-example in ANN-CBR twins, in: Case-Based Reasoning Research and Development, ICCBR 2019, Springer, Berlin, 2019, pp. 155–171.