2018 IEEE First International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)

A Machine Learning Approach to Case Adaptation

Chieh-Kang Liao, Alan Liu, Yu-Sheng Chao
Department of Electrical Engineering
National Chung Cheng University
Chiayi, Taiwan
Email: aliu@ee.ccu.edu.tw

***Abstract*—The idea of case-based reasoning (CBR) is based on experts, who prefer to rely on their experience in solving similar problems which have been solved before. Each successful experience of problem solving is stored as a case, and a case can be reused in solving a similar problem in the future. However, experience may not be exactly the same as the target problem that they are facing. To make use of experience, case adaptation is necessary. There are several issues to consider when implementing case adaptation in a CBR system, including the comprehension of each case and design of a case adaptation method. The system retrieves the most similar case depending on attributes of the target problem, and the solution part of the retrieved case will then be refined with case adaptation. The goal of this study is to design and implement case adaptation with the aide of artificial neural networks and the concept of heuristics. The experiments show that the proposed approach is efficient in adjusting the solution to fit the needs of the target problem.**

***Keywords*-case-based reasoning; case adaptation; artificial neural network; case difference heuristics**

I. Introduction

Case-based reasoning (CBR) has been used in many fields as a way of reusing the past experiences in solving the current problem [1]. As defined in [2], the CBR cycle contains four R's: Retrieve, Reuse, Revise, and Retain. The issues in CBR research include how to find the most similar case (past experience) and also how to adjust the case to fit the current needs. A major challenge is to adjust the past solution to the new problem if the new problem is not exactly the same as the past problem. Without a good method for modifying the past solution, the solution may not solve the new problem well. We have used CBR in our research on smart home management and robot control [3], [4], but our previous work relies heavily on knowledge engineers to elicit and organize data. In order to automate the process of case adaptation, we have studied the use of machine learning in CBR, and this paper reports the advancement in case adaptation.

The goal of our study in the case adaptation process is to know what to modify and how to modify it. In solving numerical solutions, we propose the use of artificial neural networks (ANN) in producing the values that can be used in adjusting solution to meet the needs of the new problem. A common way of improving CBR is to use the k nearest neighbor (kNN) method in case retrieval. The ANN then uses the retrieved cases as the training set. However, such work still possesses the problem with case adaptation.

Different approaches toward case adaptation are presented in the next section, and the rest of the paper is organized as follows. Our proposed method is explained in the third section, and the implementation and experiment evaluation are shown in Section 4, which is followed by the conclusion.

II. Case Adaptation Methods

Some depend on the work of experts, such as using decision trees and rule-based reasoning. Craw et al. use the M5' decision tree, similar to C4.5, to handle symbolic problems [5]. The internal nodes are the attributes of cases which lead to decision to the answers in the leaf nodes. Dufour-Lussier et al. propose the use of Qualitative Algebras to define rules to enhance the adaptation process [6], while Salem and Bagoury use transformational rules in case adaptation [7]. Similar to the construction of decision trees, the formation of the rules heavily relies on the work of experts.

Fuchs et al. have proposed a differential adaptation method to solve numerical problems using CBR [8]. The idea is to find the dependencies between the problem and its solution to extend to the target problem and the retrieved source problem (and its solution). The difference between the current problem and the retrieved case is calculated using a total derivative formula. This method provides a direct way of finding the adjustment of the difference. While it works well with linear problems with two parameters, it may not suite the needs of other problems.

Case-base adaptation (CBA) is proposed by Craw et al., and the main idea is to use CBR mechanism to the case adaptation process [5]. An adaptation problem consists of the new problem, the difference in the new and old problems, and the old solution. Then the case base, which consists of successful adaptation cases, is referred to find a similar adaptation case. By utilizing the retrieved adaptation case, a new solution is formed. CBA provides better flexibility to adaptation by using the concept of less over-fitting and utilizing the successful cases with minimum adjustment to the solution.

978-1-5386-9555-5/18/$31.00 ©2018 IEEE
DOI 10.1109/AIKE.2018.00023

Shiu and Pal have utilized machine learning in finding the differences in the cases and to apply the knowledge in case adaptation [9]. With the learning capability enhanced, the case maintenance has been improved. The basic concept of their approach is to form a number of clusters. Each cluster $C = \{e_1, e_2, \ldots, e_m\}$ includes m cases, in which each case $e_i = (x_{i1}, x_{i2}, \ldots, x_{in}, \theta_i)$ where $i = 1, 2, \ldots, m$ has n attributes (x) and one solution (θ). We form the attribute differences as ΔX_{ik} and the solution differences as $\Delta\theta_{ik}$. The differences of cases can be represented by the concept of discrepancy vector shown in (1).

$$\begin{aligned}\phi_{ik} &= e_i - e_k \\ &= (x_{i1} - x_{k1}, x_{i2} - x_{k2}, \ldots, x_{in} - x_{kn}, \theta_i - \theta_k) \\ &= (\Delta X_{ik}, \Delta\theta_{ik}) \end{aligned} \tag{1}$$

Our goal is to design an automated acquisition method for adaptation knowledge. For this purpose, we combine case comparison and machine learning in our approach. d'Aquin et al. propose the use of pairwise comparison among cases to find the differences, and associated rules are derived [10]. The concept of case difference heuristics (CDH) is introduced for automated generation of adaptation rules [11]. We use these methods in deriving data to train the ANN. Basically, if we have n cases, we will have n^2 adaptation rules. CDH is used in organizing those seemingly unrelated sets of data to generate a large number of data.

III. Proposed Method

The theme of our research is to reuse the retrieved case with correct modification, and we use machine learning in obtaining adaptation knowledge. The flow of the concept is depicted in Fig. 1. When a new problem (or target problem P_T) arrives, we use the attributes of P_T to perform case retrieval. After finding a similar case, the solution in that case (or retrieved solution, S_R) may be applied to the new problem after making suitable adjustment through case adaptation. The idea is to examine the difference between P_T and the retrieved problem P_R. After finding the difference, it will be used to find how much value to adjust (ΔS) in order to produce the proposed solution.

A. Case Retrieval

There are many similarity measures to use, and we use the Euclidean distance provided by a CBR tool, myCBR [12]. The formula is described as (2), in which attributes of two cases, p and q, are p_i and q_i, respectively. The distance is the square root of the sum of the squared differences between p_i and q_i. During implementation, we faced the need of magnifying the differences, and the idea of normalization is applied. We have implemented the similarity measures accordingly and validated the result with the results obtained by the tool, myCBR.

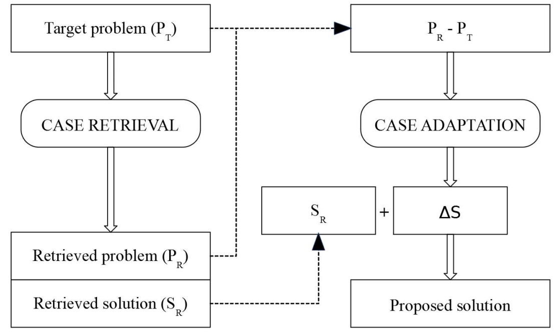

Figure 1. Proposed method.

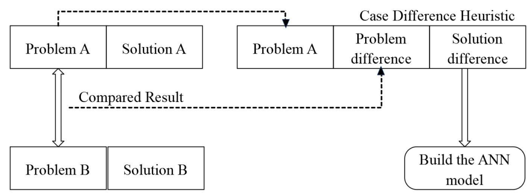

Figure 2. Case differences as the input data to an ANN.

$$d(p, q) = \sqrt{\sum_{i=1}^{n} \left(\frac{q_i - p_i}{i_{max} - i_{min}}\right)^2} \tag{2}$$

B. Use of Case Difference Heuristics

We compare cases in a pairwise manner to obtain the differences between cases and we use CDH based on the differences in the problem description and also the solution description between two cases. Fig. 2 shows the process of deriving the CDH. For a target problem (Problem B), a similar case containing the Solution A is retrieved. The differences between Problem B and Problem A and also between Solutions A and B are calculated to form CDH. All CDH entries will be used as the data to train a neural network model. In other words, an ANN model learns the problem differences and to predict the differences in the solution, which is called the prediction offset.

C. Design of an ANN Model

We have used leaky rectified linear unit, Leaky ReLU, as activation function and used TensorFlow [13] and Keras [14] in implementing the ANN model. By applying CDH in our research, we generate more cases which can be supplied to the construction of the ANN model. With more training data available, we do not require the network to be deeper. To match the number of generated attributes, we use 10 input neurons. There are 4 hidden layers with 3584 neurons in each layer. The number of neurons are obtained by gradually increasing the number of neurons until over-fitting is observed. In order to overcome a possible problem

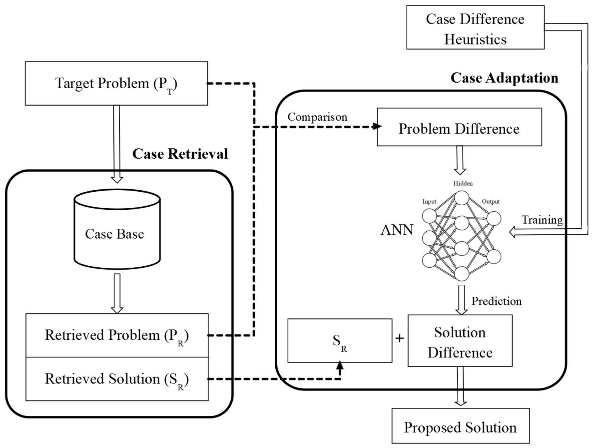

Figure 3. System Architecture.

of dead ReLU, we use Leaky ReLU as stated in the work [15].

D. Solution Prediction

Fig. 3 shows the proposed system, in which when a target problem P_T arises, the case retrieval process finds a most similar case with P_R and S_R, and the difference in P_T and P_R is compared and calculated as the problem difference. The values of all CDH are derived by pairwise comparison of all cases in the case base, leading to the construction of the ANN model. The problem difference between the current problem and retrieved problem will be entered to the ANN to produce the solution difference which will be the adjustment to the retrieved solution.

IV. Experiments

A. Dataset

We use the NACA 0012 dataset from the UCI Repository to perform experiments [16]. The purpose of this dataset was to build a prediction system for "*the self-generated noise of an airfoil blade encountering smooth flow*". Some researchers have used this dataset for predicting the noise level based on the attributes related to the wing. The data was collected from aerodynamic and acoustic tests of airfoil blade sections of NACA 0012 in a wind tunnel [17]. For each data entry, the input consists of 5 values: frequency (Hz), angle of attack (degrees), chord length (m), free-stream velocity (m/s), and suction side displacement thickness (m), while the output is scaled sound pressure level (dB). These 6 values are labeled as a1 to a6, respectively, in Table I.

B. Experiment Procedure

The procedure of the experiment follows the flow in Fig. 3, in which we divide the dataset into two portions. One portion is the case base, called CB, and the other one is the query cases, called QC. If the number of data entries in

Table I
SAMPLE DATA ENTRIES

a1 (Hz)	a2 (°)	a3 (m)	a4 (m/s)	a5 (m)	a6 (dB)
800	0	0.3048	71.3	0.002663	126.201
3150	1.5	0.3048	39.6	0.003921	119.111
4000	2	0.2286	31.7	0.003724	120.527

CB is 20% of all data, then the number of data entries in QC will be 80% of all data. If CB occupies 80% of the dataset, then QC has the remaining 20%. The results of retrieving CDH are aided as training data for constructing the ANN model.

The goal of the experiment is to use ANN to support case adaptation in CBR. To show how data size may affect the trained ANN models, we built two ANN models with different data size. We use CDH in learning case knowledge to raise the accuracy of the answer. We compare the results of CBR without case adaptation and CBR with case adaptation. The operation is as follows:

1) We randomly picked 20% of the data as the content of the initial case base. After performing pairwise comparison, we normalize the differences between the cases to obtain the CDH to train an ANN model. Another ANN model is constructed with 80% of the data.
2) After an ANN model is constructed, we use the rest of the data as the target cases to perform case retrieval. After finding the most similar case, we compare it with the target case to compute the attribute differences. The purpose is to examine the prediction ability of the ANN model.
3) Using the ANN model obtained in the step 1, we compute the difference of the obtained answer. For case adaptation, the output of the ANN is added to the answer of the retrieved case in the step 2 to produce the final answer.

C. Analysis of the results

By using the content of QC to test the performance of the CBR system, first a most similar case is retrieved containing the retrieved solution (S_R). According to the attributes given, the ANN will produce a prediction offset to add to S_R to produce a prediction solution. Table II shows a portion of the result of derived solutions. The difference of S_R and the ground truth is Error 1, while Error 2 is the difference between the prediction solution and the ground truth. We are interested in whether the value of Error 2 is smaller than Error 1.

Table III lists the mean absolute error (MAE) values obtained from the error of the retrieved solution compared to the target solution using the ANN trained with 20% of data and the ANN trained with 80% of data. Using the ANN trained with 20% of data, the MAE of the similarities of the retrieved cases with the target cases is 3.94 for performing

Table II
PORTION OF RESULTS

ground truth	retrieved solution	Error 1	prediction offset	prediction solution	Error 2
127.591	125.951	1.64	-1.277304	127.2283	0.3627
117.151	119.541	2.39	2.210737	117.3302	0.1792
124.106	126.616	2.51	2.209579	124.4064	0.3004
123.236	121.106	2.13	-1.463225	122.5692	0.6668

Table III
MAE OF CBR WITH AND WITHOUT CASE ADAPTATION

	No Adaptation	With Case Adaptation
ANN with data size 20%	3.94	2.3
ANN with data size 80%	2.14	0.84

just case retrieval; whereas the value of 2.3 is obtained for applying case adaptation. The value drops to 2.14 and 0.84, respectively when the ANN is trained with a larger set (80%) of data.

When dealing with the answer provided by the target cases, the results show that case adaptation contributed to better results. For using 20% of data to build the ANN, we have 1221 data entries to test, and 902 adapted solutions are better than the ones retrieved by the CBR without case adaptation. For the model using 80% of data, 310 entries were used for testing, and 266 answers were better than the ones retrieved without case adaptation. The t-test was conducted and the result showed that the use of case adaptation in CBR was significant.

V. CONCLUSION

Case adaptation is a difficult challenge and usually expert-dependent. In order to reduce the intervention of the experts in case adaptation, we have used machine learning techniques. This paper discusses the use of ANN to adjust the retrieved solution to fit the needs of the current target problem. Inspired by the differential method and CBA, our reason of using an ANN model in case adaptation is that the solution in the retrieved case can be adjusted by adding some numbers which are the output of the ANN model. By using CDH as the input to the ANN model, we obtained a usable ANN model producing the offset to add to the retrieved solution. Currently, this work is only suitable for numeric problems, and an in-depth study is underway for solving problems with multivariate attributes.

ACKNOWLEDGMENT

This project is partially supported by Ministry of Science and Technology, Taiwan under 106-2221-E-194-018-MY3.

REFERENCES

[1] D. W. Aha and J. Lieber, Eds., *Case-Based Reasoning Research and Development - 25th International Conference, ICCBR 2017, Trondheim, Norway, June 26-28, 2017, Proceedings*, ser. Lecture Notes in Computer Science, vol. 10339. Springer, 2017.

[2] A. Aamodt and E. Plaza, "Case-based reasoning: Foundational issues, methodological variations, and system approaches," *AI Commun.*, vol. 7, no. 1, pp. 39–59, Mar. 1994.

[3] H. Chu and A. Liu, "A negotiation method for multiple participants using case based reasoning," in *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Oct 2012, pp. 2475–2480.

[4] C.-C. Tsai, K.-S. Hwang, A. Liu, and C.-F. Juang, *Advances and Challenges on Intelligent Learning in Control Systems*. Wiley-Blackwell, 2015, ch. 6, pp. 241–264.

[5] S. Craw, N. Wiratunga, and R. C. Rowe, "Learning adaptation knowledge to improve case-based reasoning," *Artificial Intelligence*, vol. 170, no. 16, pp. 1175 – 1192, 2006.

[6] V. Dufour-Lussier, F. Le Ber, J. Lieber, and L. Martin, "Case Adaptation with Qualitative Algebras," in *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, 2013, pp. 3002–3006.

[7] A. M. Salem and B. E. Bagoury, "A case-based adaptation model for thyroid cancer diagnosis using neural networks," in *Proceedings of the Sixteenth International Florida Artificial Intelligence*, 2003, pp. 155–160.

[8] B. Fuchs, J. Lieber, A. Mille, and A. Napoli, "Differential adaptation: An operational approach to adaptation for solving numerical problems with CBR," *Knowledge-Based Systems*, vol. 68, pp. 103 – 114, 2014.

[9] S. Shiu and S. K. Pal, *Foundations of Soft Case-Based Reasoning*. USA: John Wiley & Sons, Inc., 2004.

[10] M. d'Aquin, F. Badra, S. Lafrogne, J. Lieber, A. Napoli, and L. Szathmary, "Case base mining for adaptation knowledge acquisition," in *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, 2007, pp. 750–755.

[11] V. Jalali and D. Leake, "Enhancing case-based regression with automatically-generated ensembles of adaptations," *J. Intell. Inf. Syst.*, vol. 46, no. 2, pp. 237–258, Apr. 2016.

[12] K. Bach and K.-D. Althoff, "Developing case-based reasoning applications using myCBR 3," in *Case-Based Reasoning Research and Development*, B. D. Agudo and I. Watson, Eds. Springer Berlin Heidelberg, 2012, pp. 17–31.

[13] M. Abadi *et al.*, "Tensorflow: A system for large-scale machine learning," in *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*, ser. OSDI'16, 2016, pp. 265–283.

[14] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ser. ICCV '15, 2015, pp. 1026–1034.

[16] D. Dheeru and E. Karra Taniskidou, "UCI machine learning repository," 2017, airfoil Self-Noise Data Set. [Online]. Available: http://archive.ics.uci.edu/ml

[17] T. F. Brooks, D. S. Pope, and M. A. Marcolini, "Airfoil self-noise and prediction," Tech. Rep. NASA-RP-1218, 1989.