Adaptive Case-Based Reasoning Using Support Vector Regression

Mansooreh Sharifi and Mahmoud Naghibzadeh
Department of Computer Engineering
Ferdowsi University of Mashhad
Mashhad, Iran
ma_sh200@yahoo.com, naghibzadeh@ferdowsi.um.ac.ir

Seyed Mojtaba Rouhani
Islamic Azad University of Gonabad
Gonabad, Iran
m.rouhani@ieee.org

***Abstract*— one important step in case-based reasoning systems is the adaptation phase. This paper presents a case-based reasoning system which automatically adapts past solutions to propose a solution for new problems. The proposed method for case adaptation is based on support vector regression. At first, case base is partitioned using a clustering technique. Then, a support vector regression is constructed for each cluster using local information. For solving a new problem, its local information is computed with respect to the most similar cluster and the corresponding support vector regression propose a solution. Experiment shows this approach greatly improves the accuracy of a retrieve-only CBR system with minimizing each didactic model.**

Keywords-case-based reasoning; adaptation; support vector regression; clustering

I. Introduction

Reasoning can be defined as the process by which, the new information is obtained from the data set. Generally, people solve a new problem by employing the experience of previous similar problems[1]. This experience may be obtained by themselves or by another person. With these explanations, we deduce that the case-based reasoning is similar to problem solving behavior of man. Case-based reasoning is a method of problem solving using past experiences which is a powerful and frequently applied way to solve problems for humans. In CBR terminology, a problem situation usually is known as a case. A past case is a previously experienced situation, which has been stored in the case-base and can be reused in the solving of future problems. Correspondingly, the description of a new problem to be solved is referred to as a new case or unsolved case. In general, the mechanism of a CBR system can be described as a cycle consists of four processes[2]:

1. **Retrieve**: Given a target problem, the most similar cases are retrieved from the case-base.
2. **Reuse**: In this process, the information and knowledge of retrieved cases are used for solving the target problems.
3. **Revise**: In this process, a solution is proposed for target problem.
4. **Retain**: In this process, the experiences which obtained from previous processes are stored for future problem solving.

In Fig. 1, this cycle is illustrated.

In the first step of a problem solving mechanism, the most similar cases to the new problem should be identified and retrieved. Inasmuch as we know, when two problems are similar in some degrees, they are likely to have similar responses. Thus, the retrieved items will produce a new solution. Usually, the answers need to be adapted to satisfy the new problem requirements. According to [1], case adaptation is the process by which the retrieved solution will be transformed into an appropriate solution for the current problem. Without adaptation, CBR would be a simple pattern matcher.

For acquiring adaptation knowledge, two main approaches exist[1]. In traditional approaches, the adaptation knowledge is obtained from the experience of domain experts and is coded

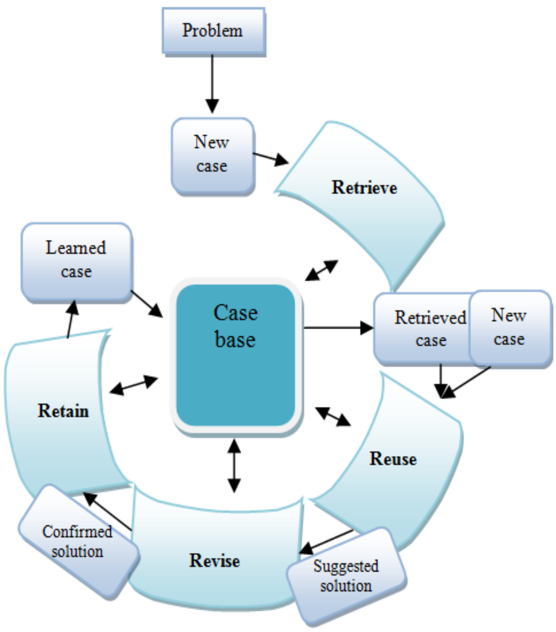

Figure 1. CBR cycle

manually into the CBR system. The representation of this knowledge could be one of the following forms: decision table, semantic tree or If–Then rules. Another approach in acquiring adaptation knowledge is using machine-learning techniques. In this approach with learning mechanisms, specialized heuristics are generated which relate the differences in the input specifications to the differences in the output specifications. Acquiring adaptation knowledge through interviews with domain experts has some drawbacks:

- Subjective and inaccurate results
- Labor intensive and time consuming
- Difficulties in maintaining the adaptation knowledge that has been acquired
- Dependency to the amount of experience of expert

For these reasons, many approaches are developed which utilize machine-learning techniques for deriving adaptation knowledge.

In this paper, we propose a method of learning adaptation knowledge for each cluster in a CBR system. Employment a separate model for each cluster in the case-base has the advantage that we can use both case-specific knowledge of past problems and additional knowledge derived from clusters of cases. With this method, the size of learning patterns for each model will be reduced and the quality of solutions will be improved. The results of experiments show that this adaptation mechanism could improve the performance of the CBR system compared to retrieval-only CBR system.

The rest of this paper is as follows: Section II reviews some existing research on adaptation process in CBR. Our adaptive CBR is described in Section III. Section IV shows the results of our experiments on several data sets. Conclusions and future works are summarized in Section V.

II. RELATED WORKS

Li et al. [3] propose a method of learning adaptation rules with pairwise comparisons of cases in the case-base. If the dissimilarity between two cases is over a predefined threshold, the adaptation rule will not generate for these cases. When all rules are generated, repetitive rules will be summarized with a certainty factor based on its frequency and others will be aggregated. Badra et al. [4] propose a method that implements a semi-automatic adaptation knowledge acquisition in adaptation phase. The idea of this work is that a set of rules are generated by calculating the differences between problem attributes and solution attributes. In adaptation step, the source and target problems are matched and an adaptation rule is selected which proposes a solution for target problem. Craw et al. [5] implement a c4.5 decision tree for learning adaptation rules. In their work, a situation/action pair is constructed for each case. Situation part consists of the differences between the new and retrieved problem while the action part contains the differences should be applied to transform the retrieved solution to an appropriate solution. After the adaptation rules are generated, a weighted voting is done for adaptation of new problems. In [6, 7] a genetic algorithm is utilized in the adaptation phase. Retrieved cases form the initial population of genetic algorithm. By applying crossover and mutation operators, genetic algorithm will produce appropriate solutions for new problems. Also Grech and Main [8] add a learning mechanism to the genetic algorithm in which a feedback from revision to reuse phase is sent, which states that genetic algorithm have mutations lead to incorrect results. This feedback mechanism prevents the spread of inappropriate values in the population. Jung et al. [9] deal with partitioning the case base. In their work, the case base is partitioned using k-means clustering technique. With representative cases, a RBF [1] neural network is constructed. For solving new problems, the most similar case to problem is sent to the network for adaptation. Zhang et al. [10] construct a RBF neural network and train it using similar cases to new problem. This RBF network propose a solution for new problem. Using retrieved cases, the error of network is calculated. By combining this error and the proposed solution provided by the RBF network, an interval solution with certain confidence can be obtained. Also some works have utilized a combination of learning methods in the adaptation phase. Policastro et al. [11] present a method which implements a set of machine learning algorithms in the adaptation phase. For each case, a set of adaptation patterns are generated by calculating the differences between the extracted case and the most similar ones. With these patterns, a set of machine learning algorithms (MLP, SVM, M5) are trained. At last, a combiner combines the outputs of these algorithms and produces the final output. Kalinkin et al. [12] present a CBR method for classification problems which trains a set of machine learning algorithms using information of adjacent cases. This information consists of: classes of neighbors, minimum distances to representatives of each class, class of nearest neighbor, etc. When the system receives a new case, the information of adjacent cases is calculated. This information will be sent to the trained model to obtain the optimal class.

CBR methods discussed in this section usually do not apply any clustering mechanism in the case management phase. This property makes them unable to benefit from the information which clustering gives us. Also those works that have utilized clustering on case management (as [9]), do not use the information derived from clusters of cases in the adaptation phase.

III. ADAPTIVE METHOD

According to explanations we illustrated in section I, in this study we propose a method for adaptation based on machine learning technique. Our adaptive method utilizes SVR [2] in adaptation phase. SVR is an advanced version of the Support Vector Machines, has been proposed for regression problems. Also, according to analysis we made in section II, utilizing a clustering technique in the management of cases in CBR can improve accuracy of the system. The general scheme of our method can be explained in the following processes:

- Case representation
- Case-base management

[1] Radial Basis Function

[2] Support vector regression

- Case adaptation

A. Case representation

The first step in designing a CBR system is selecting an appropriate structure to model cases. Since in retrieval phase, the similarity between cases is considered, selecting the suitable features is important in case representation. Generally a case structure consists of two parts: problem features and solution features. If some of the problem features are irrelevant, a feature selection mechanism must be applied to select the appropriate feature set.

B. Case-base management

After determining the case structure, the next step is how to manage the case base. Keeping all cases fully integrated has some maintenance issues. For instance, in retrieving similar cases, total of case base should be searched which is a time-consuming operation. To overcome this problem, a clustering mechanism is applied in this method. Utilization of additional information derived from clusters of cases is another advantage of using clustering techniques in case base management phase. This information will be used for training and case adaptation.

1) Case-base clustering

SOM is an artificial neural network developed by Kohonen in 1982. It has many applications in clustering purposes. After reviewing the various clustering techniques, this method uses SOM for case base clustering. According to [13], "The principal goal of the SOM is to transform an incoming signal pattern of arbitrary dimension into a one- or two-dimensional discrete map, and to perform this transformation adaptively in a topologically ordered fashion".

2) Training SVR for clusters

After partitioning the case base, for each case in the clusters, local information is computed. This information includes:

- Difference with representative case of its cluster
- Difference with the most similar case in its cluster
- Difference with the most dissimilar case in its cluster

For each cluster, we train a SVR model using its local information. With this action, each model is limited to its corresponding cluster and will propose the related solutions.

C. case adaptation

Because of the case adaptation importance, we have included the adaptation phase in our system. In this step the most similar cluster to the target problem is selected. The similarity measure is based on Euclidean distance. For a target problem, local information is computed with respect to retrieved cluster. These local information make the input vectors for corresponding SVR. Finally, the retrieved model will produce an adapted solution.

Fig. 2 illustrates the work flow of our adaptive method.

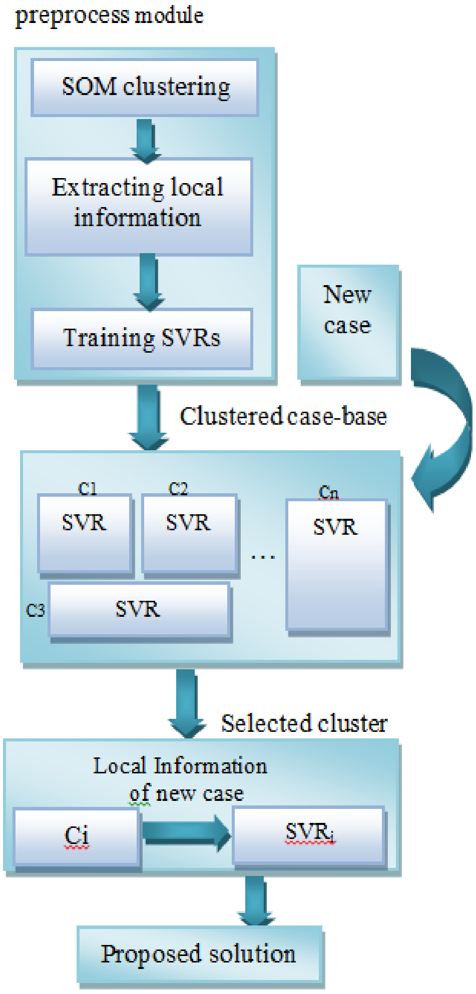

Figure 2. The work mechanism of the adaptive mehtod

In the preprocessing module, a clustering mechanism is applied which divides the case base into some partitions. Then for each case in the clusters, local information is extracted. Based on derived information, we train a separated SVR for each of them. These models will be used in the adaptation phase.

In the next module, we have a partitioned case-base with some SVRs. This module receives a new problem and retrieves the most similar cluster based on similarity measure. With respect to the retrieved cluster, local information of target problem is extracted. Finally, the corresponding SVR will propose a solution.

IV. EXPERIMENTS

For evaluating the method, two case-bases available in the Machine Learning repository of UCI[1] were used.

Servo: This data set is related to simulation of a servo system which involves a servo amplifier, a motor, a lead screw/nut, and a sliding carriage. The output value is a rise

[1] University of California, Irvine

time. In other words, the time required for the system to respond to a step change in a position set point.

Housing: This data set is related to housing values in suburbs of Boston. The purpose was to prediction of the price of a house with respect to a set of attributes.

Table I illustrates the main characteristics of these data sets.

Column #sample shows the number of cases in the original case-base. Columns #input attributes and #output attributes show the number of problem features and solution features, respectively.

The experiments were carried out using 10-fold cross validation for each case base.

TABLE II illustrates the mean absolute error for the different learning patterns in our proposed method. The local information of cases which used in training SVR are: difference with nearest neighbor in cluster, difference with hindmost neighbor in cluster, difference with representative of cluster.

The results show that using the difference of each case with its cluster representative has better consequences in solving problems.

Also for evaluation of performance, we compare our proposed method with different problem solving strategies. The first strategy is simple SVR in which a support vector regression model is trained using all cases in the case base. In this method, no local information is used in training model. The second strategy is C-CBR [1] in which, no adaptation mechanism is applied. In this strategy, a CBR system is used in which the case base has been divided using SOM technique. For solving a new problem, the most similar case in the most similar cluster will propose a solution. The third method is AC-CBR [2] which is our proposed method. Fig. 3 and Fig. 4 show the results of comparisons. In both of them, the C-CBR method has the worst consequences than others. This is due to the lack of learning mechanism in this method. Also in both figures, the AC-CBR has better consequences than simple SVR. This is due to specificity of the trained models in AC-CBR. In simple SVR, a general model is trained using all cases in the case base which causes to low quality of solutions. In the other side, in AC-CBR method a separated SVR model is trained for each cluster using local information of clusters and improves the quality of solutions. Using local information redounds to the proportion of the proposed solutions to their corresponding clusters.

TABLE I. DATA SET DESCRIPTION

Data set	# samples	# input attributes	# output attributes
Servo	167	4	1
Housing	506	13	1

[1] Clustered CBR

[2] Adaptive clustered case base reasoning

TABLE II. AVERAGE ERROR RESULTS FOR DIFFERENT LOCAL INFORMATION IN OUR PROPOSED METHOD.

Local information for training	Mean Absolute Error	
	Servo	***Housing***
Difference with nearest neighbor in cluster	0.36	2.52
Difference with hindmost neighbor in cluster	0.47	3.49
Difference with representative of cluster	**0.34**	**2.46**

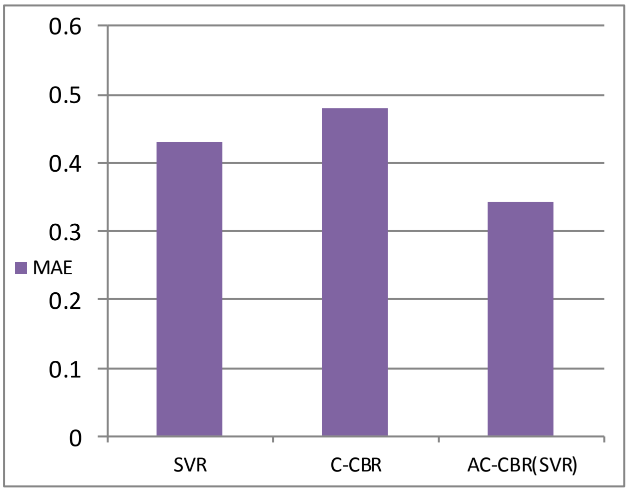

Figure 3. MAE in Servo data set

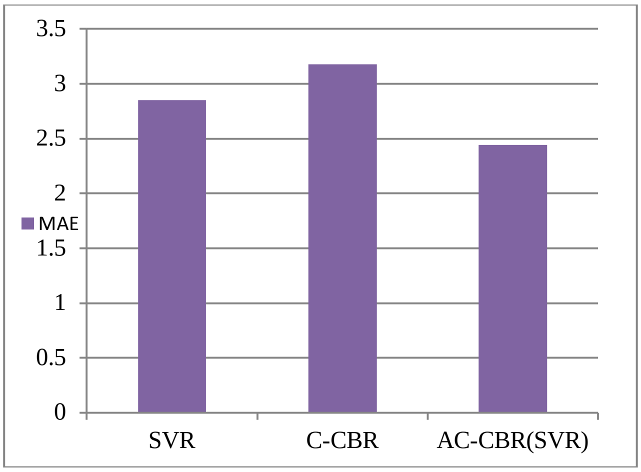

Figure 4. MAE in Housing data set

V. CONCLUSION & FUTURE WORK

In this study an adaptive CBR system was proposed which utilizes support vector regression in adaptation phase. Due to difficulties in maintaining case-base fully integrated, the SOM clustering technique was utilized. Also for using clustering benefits the information derived from clustering was employed in adaptation phase. The experiments show this method can truly improve the accuracy of a retrieval only CBR system.

The focus of this study was on data sets which have one solution feature in their attributes. In the future we decide to extend this work for data sets which have more attributes in their solutions by training a separate model for each component of the solution. Another extension to this work is storing the knowledge obtained from solving new problem in the case-base. However this idea is a good solution for improving accuracy, but causes maintenance and update difficulties.

REFERENCES

[1] S. Shiu and S. K. Pal, *Foundations of Soft Case-Based Reasoning*. Hoboken, New Jersey: John Wiley & Sons 2004.

[2] A. Aamodt and E. Plaza, "Case-Based Reasoning:Foundational Issues, Methodological Variations, and System Approaches", Artificial Intelligence Communications, vol. 7, pp. 39-59, 1994.

[3] H. Li, D. Hu, T. Hao, L. Wenyin, and X. Chen, "Adaptation Rule Learning for Case-Based Reasoning", presented at the Semantics, Knowledge and Grid, 2009.

[4] F. Badra, J. Lieber, and E. Nauer, "Using Knowledge Discovery Techniques to Acquire Adaptation Knowledge in Case-Based Reasoning", in *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, 2009.

[5] S. Craw, N. Wiratunga, and R. C. Rowe, "Learning Adaptation Knowledge to Improve Case-Based Reasoning", *Artificial Intelligence,* vol. 170, pp. 1175-1192, 2006.

[6] S. Passonea, P. W. H. Chungb, and V. Nassehi, "Incorporating domain-specific knowledge into a genetic algorithm to implement case-based reasoning adaptation", *Knowledge-Based Systems* vol. 19, pp. 192-201, 2006.

[7] Y. K. Juan, S. G. Shih, and Y. H. Perng, "Decision support for housing customization: A hybrid approach using case-based reasoning and genetic algorithm", *Expert Systems with Applications,* vol. 31, pp. 83-93, 2006.

[8] A. Grech and J. Main, "Case-Base Injection Schemes to Case Adaptation Using Genetic Algorithms", presented at the Advances in case based reasoning, 2004.

[9] S. Jung, T. Lim, and D. Kim, "Integrating radial basis function networks with case-based reasoning for product design", *Expert systems with Applications,* vol. 36, pp. 5695-5701, 2009.

[10] F. Zhang, M. H. Ha, X. Z. Wang, and X. H. Li, "CASE ADAPTATION USING ESTIMATORS OF NEURAL NETWORK", presented at the Third International Conference on Machine Leaming and Cybernetics, 2004.

[11] C. A. Policastro, A. C. Carvalho, and A. C. Delbem, "Automatic knowledge learning and case adaptation with a hybrid committee approach", *JOURNAL OF APPLIED LOGIC,* vol. 4, pp. 26-38, 2006.

[12] M. Kalinkin, A. Loginov, B. Lang, and B. Novikov, "Different strategies to learning case adaptation knowledge in classification problems", 2009.

[13] S. Haykin, *Neural Networks A comprehensive foundation*. singapore: Prentice Hall, 1999.