2012 10th International Conference on Frontiers of Information Technology

Optimization of Decision Making in CBR Based Self-Healing Systems

Saadia Nasir, Maria Taimoor, Hina Gul, Amna Ali
Department of Computer Science
Kinnaird College for Women
Lahore, Pakistan
nasir.saadia@gmail.com

Malik Jahan Khan
Department of Computer Science
Namal College
Mianwali, Pakistan
malik.jahan@namal.edu.pk

***Abstract*— Autonomic systems are the software systems capable to manage themselves. These systems undergo a learning process to achieve this capability. Case-based reasoning (CBR) is one of the promising learning paradigms for autonomic managers. Autonomic managers monitor the pulse of the monitored system on periodic basis and analyze the captured state of the system. In case of a problematic state, autonomic managers use their CBR based decision support system to rectify the problem. One of the critical problems in such systems is recovery from failures. The problem of identifying the factors affecting the performance of CBR system is a key element to build successful and accurate decision support systems. For this purpose, a hybrid CBR based self-healing system supported by attribute selection methods has been proposed. An empirical investigation has been conducted in this paper using different similarity measures, solution adaptation methods and attribute selection techniques. To address the performance problem of CBR in self-healing systems, we have conducted experiments on an emulator of self-healing systems called RUIBiS using different machine learning techniques to determine the significance of weights for these similarity distances.**

***Index Terms*—Self-healing system, case-based reasoning, attribute ranking methods, performance improvement.**

I. INTRODUCTION

After the advent of internet, computing systems management is now not only restricted to managing individual software systems, but requires integrating together several heterogeneous environments. Although the cost of IT equipment has decreased, yet the cost of computing has increased. Skilled personnel are expensive to hire and approximately 33-50% of a company's budget is invested in preventing and recovering from crashes [1]. The complexity has increased the cost and errors of managing IT systems. Major problems are being faced in the areas of cost, availability and user experience. Managing a large number of computing devices connected over the global network can be a nightmare for a human administrator, no matter. New programming languages have made possible the development of large and complex systems. However, such innovations will not help administrators in dealing with system complexity. It is becoming impossible to anticipate all component complexities and these are left to be dealt at run time [2], [3].

Autonomic computing paradigm enables a system to manage itself and dynamically adapt to the changes in their environment [1], [2], [3]. Thus, no human intervention is required. Systems can adapt themselves according to the changing conditions based on the knowledge available from business policies and objectives [2]. An autonomic system has four self-management properties: self-configure, self-heal, self-optimize and self-protect [1].

Case Based Reasoning (CBR) [4], [5], [6] is a machine learning approach which enables learning in decision support systems based on past experience. Experience is stored in the form of problem-solution pairs known as cases. Nearest neighborhood is utilized to compute solution of a new problem. It is a continuous learning paradigm as new solutions along with problem information are added to its experience repository called case-base. In CBR methodology a newly formulated solution is immediately available for reuse.

CBR has been used as one of the attractive options as the decision support method inside self-healing systems [7], [8]. Autonomic managers sense the problems in the underlying monitored resource periodically. On observing an unstable state in the monitored system, CBR engine is utilized as the core of decision support system inside autonomic managers. The engine maintains a case-base of autonomic problems along with solutions inside its case-base. It compares the newly monitored problem with all of the existing problems and retrieves a set of nearest neighbors using a similarity measure. Then a solution algorithm is exploited to compute solution of the monitored problem through reusing the solutions of nearest neighbors. The decided solution is handed over to autonomic manager. Autonomic manager applies the solution onto the monitored resource to automatically and autonomously rectify the problem.

This research work focuses on the optimization of the CBR based decision making process inside self-healing systems. In [7] and [8], the conventional CBR has been exploited in autonomic systems. This work extends the CBR based autonomic manager in order to improve their performance. Standard algorithms of attribute selection have been explored and combined with CBR engine in order to rank the attributes of cases, so that the contribution of each attribute in similarity computation is justified. As part of the performance optimization, accuracy based brute-force search has been

978-0-7695-4927-9/12 $26.00 © 2012 IEEE
DOI 10.1109/FIT.2012.21

applied to find optimal cardinality of nearest neighborhood. The proposed approach has been applied on a CBR based emulator of an auction prototype RUBiS [8]. The results show that the proposed approach has improved performance as compared to the conventional CBR system.

Rest of the paper is organized as: section II presents the related literature review, section III outlines the proposed approach, section IV discusses the implementation and results and finally section V concludes the paper.

II. BACKGROUND AND RELATED WORK

Autonomic manager [1], [2], [3] performs self-managing task in a continuous control loop known as MAPE cycle: Monitor, Analyze, Plan and Execute. .CBR cycle covers the analysis and planning phases [7], [8]. For monitoring, there exists an external component that monitors the managed element through sensors. The sensors judge the state of the managed element and compare it with the target state. If the deviation between the two states exceeds a certain tolerance limit, then the monitor prepares a case and hands it over to the retrieve phase of the CBR engine. The analyze phase covers the retrieval process. The current case is then compared with existing cases and the set of most similar cases is identified. Different similarity measures are used to find the similar cases. The current case and the identified set are then handed over to the plan phase. In this phase the decisions for retrieval or reuse and retention are taken. The solution is adapted to fit the problem context. Finally the solution is applied in the execute phase.

CBR has been widely used on other domains as well. Medes and Mosley [9] have compared the prediction accuracy of eight CBR techniques in CBR based system for web hypermedia. One technique amongst these was similarity measures. In the context of this research, three similarity measures, un-weighted Euclidian Distance, weighted Euclidean distance, and Maximum measure, were investigated. Two separate approaches were introduced for weight assignment. One was on the basis of statistically significant correlation. Secondly, the linear association between the predictors and response variables by using a one-tailed Pearson's correlation was used. Their coefficient values were used as weights.

Nunez et al. [10] conducted a comparative study of most commonly used un-weighted similarity measures for classification tasks. They also proposed a weight-sensitive L'Eixampledistance similarity measure to retrieve cases. The drawback of L'Eixampledistance was a local weighting scheme. They did not automate assignment of weights.

Ji [11] conducted a research for cost estimation that is based on Euclidian distance based similarity measuring solution for CBR systems. They used genetic algorithm to assign weight values to attributes.

Dong [12] focuses on the influence of similarity measures on their proposed system's performance that is to solve the real credit assessment problems of the company. They used weighted Manhattan distance and Euclidean distance. They calculated weights by using linear regression and multivariate discriminant analysis.

III. OPTIMIZATION OF CBR BASED SELF-HEALING SYSTEMS

The existing case based autonomic systems had certain limitations such as:

- The retrieval depends upon the k-nearest neighbors measure. The retrieval is accurate depending upon the similarity measure being used.
- Human expert is required to decide which solutions to apply and then retain in the case-base.
- The significance of the attributes is difficult to decide and requires human expertise to make a decision.

This paper addresses the limitations in the following way:

- As nearest neighbor is an important factor in determining the solution to the problem, optimal k-nearest neighbor algorithm based on brute-force search has been proposed to determines the optimal nearest neighborhood cardinality. This improves the efficiency of the system, as the case-base repository grows in case based systems.
- Each attribute has a relative significance. A weighted attribute ranking within CBR engine has been proposed.

A. Algorithm 1: Finding Optimal Nearest Neighborhood Cardinality

Algorithm 1 takes into account the training case-base A of size n, the similarity measure SM to be applied on A, the value of the nearest neighbor k, and the test case-base B which is to be compared with A to find a set of closest matching solution pairs.

The cardinality (k) of the nearest neighbor has a definite impact on the performance of overall system in terms of accuracy as well as efficiency. The algorithm outlined in Fig. 1 is proposed to find an optimal value of k that will yield the highest accuracy. The value of k can be varied from 1 to the size of case-base.

B. Algorithm 2: Attribute Selection Mechanism for CBR Engine

This algorithm is outlined in Fig. 2 and aims at finding the relative importance of different attributes on the basis of the optimal value of k. It calculates the significance of each attribute. The significance measure is used as weight for calculating the similarity measure value of the respective attribute in the case. This weighted similarity measure improves the retrieval as it assigns the highest weight to the most significant attribute. We propose an empirical approach to pick the appropriate attribute selection mechanism.

The input to this algorithm is the training case-base A containing n cases and pool of attribute selection algorithms. Each attribute selection method is applied to A and a weight vector WV containing the significance of each parameter P is calculated. WV is compute for all algorithms belonging to the pool and the output with highest accuracy is picked.

Input: Training case-base A of size n, test case-base B, similarity measure SM
Output: Optimal cardinality of nearest neighborhood k_{opt}
Method:
1. optimalAcc = 0
2. For k=1 to n do
   a. For each case $x_i \in B$ do
      i. For each case $x_j \in A$ do
         sim[j] = FindSimilarity(x_i, x_j)
      ii. End For
      iii. Find sol[i] using k-NN
      iv. currentAcc = Update overall accuracy
   b. End For
   c. If currentAcc > optimalAcc Then
      i. $k_{opt} = k$
      ii. optimalAcc = currentAcc
   d. End If
3. End For
4. Return k_{opt}

Fig. 1. Finding optimal cardinality of nearest neighborhood

Input: Training case-base A containing n cases
Output: Best weight vector BWV containing significance sig_m of each parameter P_m
Method:
1. For each RA $\in$ Pool of ranking algorithms do
   a. Run RA on A
   b. Compute WV_{RA}
   c. WV_{RA} = normalize(WV_{RA})
   d. Run Algorithm 1 using WV_{RA} and compute performance Acc_{RA}
2. End For
3. BWV = selectBestRA(Acc_{RA})
4. Return BWV

Fig. 2. Finding optimal attribute selection algorithm for CBR engine

If the proposed solution is not appropriate, then the solution is adapted to find a better solution. This adaptation can be carried out any number of times with the tradeoff of computational time.

IV. Implementation and Results

A. Case Study: Rice University Bidding System (RUBiS)

The experiments have been conducted on Rice University Bidding System (RUBiS) [13]. RUBiS is an auction site prototype modeling the fundamental auction functionalities. The main functionalities include registration, browsing, bidding and selling and it is used as a benchmark for evaluating the design patterns, application server's scalability and performance. In order to inject the autonomic behavior in RUBiS, CBR based autonomic framework called CaseBasedAutonome (CBA) has been developed and used in [8]. The experiments conducted in this research use the same model of CBA.

An emulator (RUBiS user actions), a service monitor and a problem resolution manager have been embedded with RUBiS as shown in Fig. 3.

An externalized autonomic system comprising of an emulator, a service monitor and a problem resolution manager was designed. Emulator imitates the user actions which characterize the system state. The system state is periodically monitored by the service monitor and the problem resolution manager executes the planned remedial action. The case-base has been generated using the emulator. The case-base contains problems regarding the configuration and healing problems of the system. The suitable solution is suggested by the CBA, an autonomic solution finder, on encountering a problem and the planned solution is executed by the problem resolution manager.

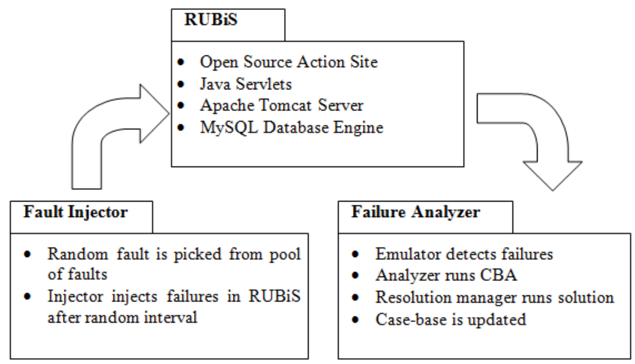

Fig. 3. CBA architecture embedded with RUBiS

B. Case Representation

In CBA, each case is represented by a set of seven parameters: Service name (SN), Permissions on the user table of the database (PU), Permissions on the item table of the database (PI), Permissions on the bid table of the database (PB), Apache Server Configuration (SC), Apache Server Health (SH), and MySQL Server Health (MH).All these are binary variables. The eighth parameter is the Configuration Solution Script (CSS) to be executed by the problem resolution manager. This parameter is a nominal variable. Each case of the RUBiS is represented as a vector of above mentioned parameters:

C_i = {SN, PU, PI, PB, SC, AH, MH, CS}

C. Empirical Evaluation of CBR Cycle

Three common similarity measures have been used for the retrieval of the nearest neighbors. The measures used are Euclidean Distance, Hamming Distance and Squared Cord Distance. The analysis presents the comparison of the performance of these similarity measures on the basis of the results of accuracy and root mean squared error (RMSE) against varying number of nearest neighbors.

1) Solution Algorithms

The solution of the nearest neighbors is aggregated using a solution algorithm, to devise a solution in the reuse phase. Weighted arithmetic average is used in the RUBiS case study.

2) Retrieval Strategy

Each attribute in the case-base has relative importance. Five algorithms have been used to judge the relative significance of each attribute. These algorithms are Information Gain, Gain Ratio, Chi Squared, OneR and Symmetrical Uncertain Attribute Evaluator. The results of these algorithms have been used as attribute weights in the retrieval algorithm.

3) Revision Strategy

If the desired solution has not been feasible, the solutions of the nearest neighbor are applied iteratively until a specified neighbor fixes up the problem.

4) Cardinality of Nearest Neighbors

The solution algorithm requires the set of nearest neighbors for formulating a solution. The cardinality can be varied from 1 to size of case-base. The purpose is to find the similarity measure that gives the optimal results with minimal cardinality.

5) Hold-Out Based Cross Validation

In this method, the original sample is randomly partitioned into m subsamples. Of the m subsamples, one of the samples is retained as validation data for testing purpose and the remaining m – 1 subsamples are used as training data set. Then cross-validation process is applied m times with each subsample used once as validation data. The m results are then averaged out to produce a single estimation.

In our case, m is equal to 1, so a single sample has been used as validation dataset.

6) Performance Measures

Accuracy and RMSE are used to empirically evaluate the performance of the proposed solutions.

$$Accuracy = \frac{Total\ correct\ prediction}{Total\ Cases} * 100$$

$$RMSE\ (x_1, x_2) = \sqrt{\frac{\sum_{i=1}^{n} (x_{1,i} - x_{2,i})^2}{n}}$$

D. Results and Discussion

1) Experiment 1: Optimal Cardinality of Nearest Neighborhood

This experiment has been conducted to find the optimal cardinality of nearest neighborhood that can give the highest accuracy rate. The value of k has been varied from 1 to 6. If the value of k is increased the accuracy rate decreases and the RMSE increases as shown in Fig. 4 and Table I. The accuracy results showed that the value of k = 1 has the highest accuracy rate. So, the empirical investigation proved that k = 1 is the optimal number of nearest neighbors to be considered in RUBiS.

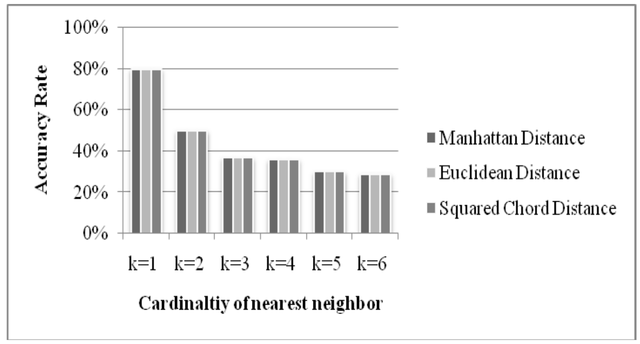

Fig. 4. Accuracy of the optimal value of nearest neighbor experiment

TABLE I. RMSE OF EXPERIMENT 1

Similarity Measure	k=1	k=2	k=3	k=4	k=5	k=6
Manhattan Distance	0.447	0.707	0.794	0.799	0.837	0.843
Euclidean Distance	0.447	0.707	0.794	0.799	0.837	0.843
Squared Chord Distance	0.447	0.707	0.794	0.799	0.837	0.843

2) Experiment 2: Optimal Ranking Algorithm

On the basis of the results of experiment 1, this empirical investigation has been conducted to find the optimal ranking algorithm. The results of this experiment revealed that Gain Ratio and Symmetrical Uncertain Evaluators are the optimal ranking algorithms that achieved the highest accuracy rate with k = 1, as shown in Fig. 5 and Table II. 70% accuracy rate has been achieved with Gain Ratio and 69% has been achieved with Symmetrical Uncertain Evaluator. The reason for better performance of these two algorithms is that they equally divide the data.

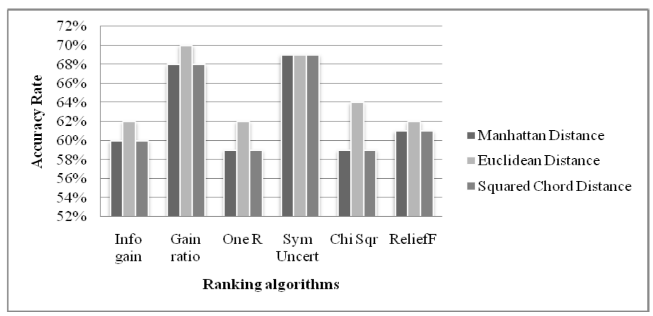

Fig. 5. Accuracy of optimal ranking algorithm

TABLE II. RMSE OF EXPERIMENT 2

Similarity Measures	Info Gain	Gain Ratio	One R	Sym Uncert	Chi Sqr	ReliefF
Manhattan Distance	0.632	0.566	0.640	0.557	0.640	0.624
Euclidean Distance	0.616	0.566	0.616	0.557	0.600	0.616
Squared Chord Distance	0.632	0.566	0.640	0.557	0.640	0.624

3) Experiment 3: Optimal Adaptation

The results of experiments 1 and 2 have been used to conduct experiment 3.The objective of experiment 3 has been to investigate the effectiveness of revising the solution if the first solution has been inappropriate. According to experiment 2, Gain Ratio and Symmetrical Uncertain Evaluators are the optimal ranking algorithms. These two ranking algorithms are used in assessing the effectiveness of revising the solution. The iterative adaptation strategy revealed inspiring results. The results of this experiment revealed that 99% accuracy rate can be achieved by revising the solution once as shown in Fig. 6 and Table III. The Accuracy rate increased and the RMSE. This shows that the second next solution can also be considered as a best solution. The only tradeoff of this experiment is between performance efficiency and accuracy.

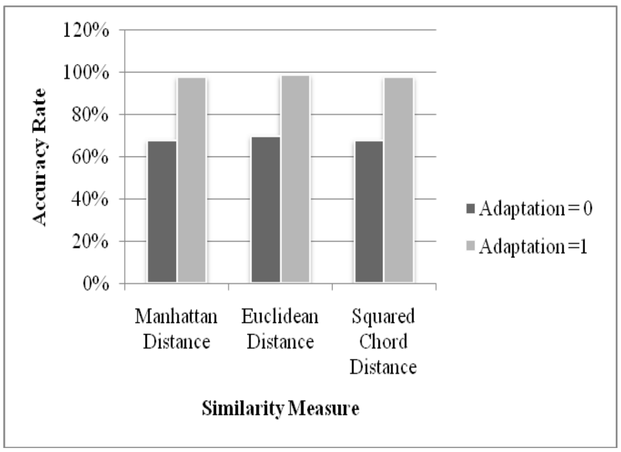

Fig. 6. Accuracy result of Gain Ratio

TABLE III. RMSE OF GAIN RATIO

Similarity Measure	Adaptation = 0	Adaptation =1
Manhattan Distance	0.5656	0.1414
Euclidean Distance	0.5656	0.1414
Squared Chord Distance	0.5656	0.1414

V. CONCLUSION

The results of the experiments revealed that the performance of the case-based autonomic systems has been improved in terms of accuracy to 99%.It has been revealed that the cardinality of one gave the highest accuracy therefore the immediate nearest neighbor is the most suitable of all the candidate solutions.

Gain Ratio and Symmetrical Uncertain Attribute Evaluators have performed as the better attribute selection mechanisms for CBR engine used inside self-healing systems as compared to other alternates. One main reason for their better performance is that they divided the dataset equally and classify it according to a single attribute or a class respectively.

Simple revision strategy of picking second best solution works well and improves the performance. The overall accuracy rate achieved has gone up to 99%. Although there is a tradeoff between efficiency and accuracy rate achieved.

REFERENCES

[1] J. O. Kephart, D. M. Chess, "Vision of Autonomic Computing", IEEE Computer Society, 36, 41–50.

[2] D. Patterson et al, "Recovery oriented computing (roc): Motivation, definition, techniques, and case Studies", UC Berkeley CS Tech. Rep. UCB/CSD-02-1175, March 2002.

[3] IBM Manifesto, "Autonomic Computing: IBM's perspective on the state of technology", IBM, 2001.

[4] A. Aamodt and E. Plaza, "Case-based reasoning: Foundational issues, methodological variations, and system approaches," in AI Communications, vol. 7:1, pp. 39–59, IOS Press, March 1994.

[5] P. Cunningham, "CBR: Strengths and weaknesses," in Proceedings of 11th Int. Conf. on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, pp. 517–523, Springer Verlag, 1998.

[6] I. Watson and F. Marir, "Case-based reasoning: A review," The Knowledge Engineering Review, vol. 9, no. 4, pp. 327–354, 1994.

[7] S. Montani and C. Anglano, "Achieving self-healing in service delivery software systems by means of case-based reasoning," Applied Intelligence, Springer, 2007.

[8] M. J. Khan, M. M. Awais, S. Shamail, I. Awan, "An Empirical Study of Enabling Self-Management Capabilities in Autonomic Systems using Case-based Reasoning", Simulation Modelling Practice and Theory, vol. 19, no. 10, November 2011, Elsevier.

[9] E. Mendes and N. Mosley, "Further Investigation into the Use of CBR and Stepwise Regression to Predict Development Effort for Web Hypermedia Applications", Proceedings of Internation Symposium on Empirical Software Engineering, 2002.

[10] H. Núñez, M. Sànchez-Marrè, U. Cortés et al., "A comparative study on the use of similarity measures in case-based reasoning to improve the classification of environmental system situations", Environmental Modelling and Software, vol. 19, no. 9, 2004.

[11] S. Ji, M. Park, H. Lee and Y. Yoon, "Similarity measurement method of case-based reasoning for conceptual cost estimation", Proceedings of International Conference on Computing in Civil and Building Engineering, 2010.

[12] Y. Dong, ""A Case Based Reasoning System for Customer Credit Scoring: Comparative Study of Similarity Measure", Proceedings of 51st Annual Meeting of the International Society for the Systems Sciences, 2007.

[13] "RUBIS: Rice university bidding system," http://rubis.objectweb.org/, Accessed on August 12, 2012.