Applied Intelligence 21, 233–238, 2004

Case-Based Reasoning: Concepts, Features and Soft Computing

SIMON C.K. SHIU
Department of Computing, Hong Kong Polytechnic University, Hong Kong, P.R. China

SANKAR K. PAL
Machine Intelligence Unit, Indian Statistical Institute, Calcutta, India

Abstract. Here we first describe the concepts, components and features of CBR. The feasibility and merits of using CBR for problem solving is then explained. This is followed by a description of the relevance of soft computing tools to CBR. In particular, some of the tasks in the four REs, namely Retrieve, Reuse, Revise and Retain, of the CBR cycle that have relevance as prospective candidates for soft computing applications are explained.

Keywords: merits of case-based reasoning, soft case-based reasoning

1. What is CBR?

The field of Case-Based Reasoning (CBR), which has a relatively young history, arose out of the research in cognitive science. It was focused on problems such as: how people learn a new skill, and how humans generate hypotheses about new situations based on their past experiences. A typical example of using CBR is medical diagnosis. When faced with a new patient, the doctor examines the patient's current symptoms, and compares with those patients that were having similar symptoms before. The treatments of those similar patients are then used and modified, if necessary, to suit the current new patient, i.e., some adaptation to the previous treatments is needed. In real life there are many such similar situations which employ this CBR paradigm to build reasoning systems, such as retrieving preceding law cases for legal arguments; determining the house prices based on similar information from other real estates; forecasting weather conditions based on previous weather records, and synthesizing a material production schedule from the previous plans.

In most CBR systems, the internal structure can be divided into two major parts: the case retriever and the case reasoner, as shown in Fig. 1. The case retriever's task is to find the appropriate cases in the case base, while the case reasoner uses the retrieved cases to find a solution to the given problem description. This reasoning process generally involves both determining the differences between the retrieved cases and the current query case, and modifying the retrieved solution to appropriately reflect these differences. The reasoning process may, or may not, involve retrieving further cases or portions of cases from the case base.

The problem solving life cycle in a CBR system consists essentially of four parts (i.e., the four REs): (i) *Retrieving* similar previously experienced cases whose problem is judged to be similar; (ii) *Reusing* the cases by copying or integrating the solutions from the cases retrieved; (iii) *Revising* or adapting the solution(s) retrieved in an attempt to solve the new problem; and (iv) *Retaining* the new solution once it has been confirmed or validated.

The idea of CBR is intuitively appealing because it is similar to human problem solving behavior. People draw on past experience while solving new problems, and this approach is both convenient and effective, and it often relieves the burden of in-depth analysis of the problem domain. This leads to the advantage that CBR can be based upon shallow knowledge and does not require significant effort in knowledge engineering when compared with other approaches (e.g., rule-based).

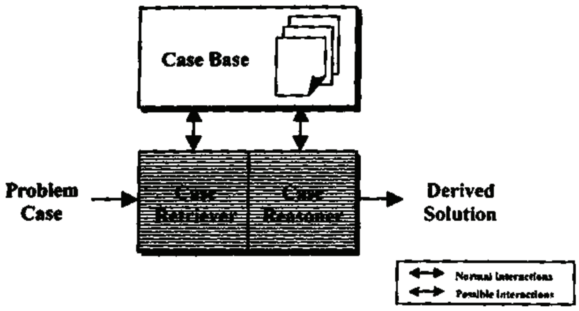

Figure 1. Two major components of a CBR system.

2. Where to Use and Why?

Although CBR is useful for various types of problems and domains, there are situations when it is not the most appropriate methodology to employ. There are a number of characteristics of candidate problems and their domains, as mentioned below, that determine the applicability of CBR [1–3]:

- The domain doesn't have an underlying model.
- There are exceptions and novel cases.
- Cases recur frequently.
- There is significant benefit in adapting past solutions.
- Relevant previous cases are obtainable.

In general, there are a number of merits of using CBR:

Reduce the knowledge acquisition task. By eliminating the need of extraction of a model, or a set of rules, as is necessary in model/rule-based systems, the knowledge acquisition tasks of CBR consist mainly of the collection of the relevant existing experiences/cases and their representation and storage.

Avoid repeating mistakes made in the past. In systems that record failures as well as successes, and perhaps the reason for those failures, the information about what caused failures in the past can be used to predict potential failures in the future.

Provide flexibility in knowledge modeling. Model-based systems, due to their rigidity in the problem formulation and modeling, sometimes cannot solve a problem which is on the boundaries of their knowledge or scope, or when there is some missing or incomplete data. In contrast, case-based systems use the past experiences as the domain knowledge and can often provide a reasonable solution, through appropriate adaptation, to these types of problems.

Reason in domains that have not been fully understood, defined or modeled. In situation where insufficient knowledge exists to build a causal model of a domain or to derive a set of heuristics for it, a case-based reasoner can still be developed using only a small set of cases from the domain. The underlying theory of the domain knowledge does not have to be quantified or understood entirely for a case-based reasoner to function.

Make predictions of the probable success of a proffered solution. When information is stored regarding the level of success of past solutions, the case-based reasoner may be able to predict the success of the suggested solution to a current query problem. This is done by referring to both the stored solutions, the level of success of these solutions, and the differences between the previous and current contexts of applying these solutions.

Learn over time. As CBR systems are used, they encounter more problem situations and create more solutions. If solution cases are subsequently tested in the real world, and a level of success is determined for those solutions, then these cases can be added into the case base, and used to help solving future problems. As cases are added, a CBR system should be able to reason in a wider variety of situations, and with a higher degree of refinement and success.

Reason in a domain with a small body of knowledge. While in a problem domain for which there is only a few cases available, a case-based reasoner can start with these few known cases and incrementally build its knowledge as cases are added. The addition of new cases will cause the system to expand in directions that are determined by the cases encountered in its problem solving endeavors.

Reason with incomplete or imprecise data and concept. As cases are retrieved, they may not be identical to the current query case. Nevertheless, when they are within some defined measure of similarity to the query case, any incompleteness and imprecision can be dealt with by a case-based reasoner. While these factors may cause a slight degradation in performance, due to the increased disparity between the current and retrieved cases, reasoning can still continue.

Avoid repeating all the steps that need to be taken to arrive at a solution. In problem domains that require significant processes to create a solution from scratch, the alternative approach of modifying a previous solution can significantly reduce this processing requirement. In addition, reusing a previous solution also allows the actual steps taken to reach that solution to be reused for solving other problems.

Provide a means of explanation. Case-based reasoning systems can supply a previous case and its (successful) solution to help convince a user, or to justify the reasons, regarding why a proposed solution to their current problem should be considered. In most domains, there will be occasions when a user wishes to be reassured about the quality of the solution provided by a system. By explaining how a previous case was successful in a situation, using the similarities between the cases and the reasoning involved in adaptation, a CBR system can explain its solution to a user. Even for a hybrid system, one that may be using multiple methods to find a solution, this proposed explanation mechanism can augment the causal (or other) explanation given to the user.

Can be used in many different ways. The number of ways a CBR system can be implemented is almost unlimited. It can be used for many purposes; a few examples are: creating a plan, making a diagnosis, and arguing a point of view. Therefore the data dealt with by a CBR system is likewise able to take many forms, and the retrieval and adaptation methods will also vary. Whenever stored past cases are being retrieved and adapted, case-based reasoning is said to be taking place.

Can be applied to a broad range of domains. Case-based reasoning can be applied to extremely diverse application domains. This is due to the seemingly limitless number of ways of representing, indexing, retrieving and adapting cases.

Reflect human reasoning. As there are many situations where we, as humans, use a form of case-based reasoning, it is not difficult to convince implementers, users and managers of the validity of the paradigm. Likewise, humans can understand a CBR system's reasoning and explanations, and are able to be convinced of the validity of the solutions they receive from a system. If a human user is wary of the validity of a received solution, they are less likely to use this solution. The more critical the domain, the lower the chance a received solution will be used, and the greater the required level of a user's understanding and credulity.

3. Soft Case-Based Reasoning

3.1. What is Soft Computing?

Soft computing, according to Lotfi Zadeh [4], is "an emerging approach to computing, which parallels the remarkable ability of the human mind to reason and learn in an environment of uncertainty and imprecision." In general, it is a consortium of computing tools and techniques, shared by closely related disciplines including fuzzy logic (FL), neural network theory (NN), evolutionary computing (EC) and probabilistic reasoning (PR); with the latter discipline subsuming belief networks, chaos theory and parts of learning theory. Recently, the development of rough set theory by Zdzislaw Pawlak [5] adds a further tool for dealing with vagueness and uncertainty arising from granulation in the domain of discourse. In soft computing, the individual tool may be used independently depending on the application domains. They can also act synergistically, not competitively, for enhancing the application domain of the other by integrating their individual merits, e.g., the uncertainty handling capability of fuzzy sets, learning capability of artificial neural networks, and the robust searching and optimization characteristics of genetic algorithms. The primary objective is to provide flexible information processing systems that can exploit a tolerance for imprecision, uncertainty, approximate reasoning, and partial truth, in order to achieve tractability, robustness, low solution cost and closer resemblance to human decision making.

The notion of fuzzy sets was introduced by Zadeh in 1965. It provides an approximate but effective and flexible way of representing, manipulating and utilizing vaguely defined data and information. It can also describe the behaviors of systems that are too complex, or too ill-defined, to allow precise mathematical analysis using classical methods and tools. Unlike conventional sets, fuzzy sets include all elements of the universal set but with different membership values in the interval [0, 1]. Similarly, in fuzzy logic, the assumption upon which a proposition is either true or false is extended into multiple value logic, which can be interpreted as a degree of truth. The primary focus of fuzzy logic is on natural language where it can provide a foundation for approximate reasoning using words (i.e., linguistic variables).

Artificial neural network models are attempts to emulate electronically the architecture and information representation scheme of biological neural networks.

The collective computational abilities of the densely interconnected nodes or processors may provide a natural technique in a manner analogous to humans. Neuro-fuzzy computing, capturing the merits of fuzzy set theory and artificial neural networks, constitutes one of the best-known hybridizations in soft computing. This hybrid integration promises to provide, to a greater extent, more intelligent systems (in terms of parallelism, fault tolerance, adaptivity, and uncertainty management) able to handle real life ambiguous recognition or decision-making problems.

Evolutionary Computing (EC) describes adaptive techniques, which are used to solve search and optimization problems, inspired by the biological principles of natural selection and genetics. In EC, each individual is represented as a string of binary values; populations of competing individuals evolve over many generations according to some fitness function. A new generation is produced by selecting the best individuals and *mating* them to produce a new set of offspring. After many generations, the offspring contain all the most promising characteristics of a potential solution for the search problem.

Probabilistic computing has provided many useful techniques for the formalization of reasoning under uncertainty, in particular the Bayesian and belief functions, and the Dempster-Shafer theory of evidence. The rough set approach deals mainly with the classification of data and synthesizing an approximation of particular concepts. It is also used to construct models that represent the underlying domain theory from a set of data. Often, in real life situations, it is impossible to define a concept in a crisp manner. For example, given a specific object, it may not be possible to know to which particular class it belongs, the best knowledge derived from past experience may only give us enough information to conclude that this object belongs to a boundary between certain classes. The formulation of the lower and upper set approximations can be generalized to some arbitrary level of precision, which forms the basis for rough concept approximations.

3.2. *Why Soft Computing in CBR?*

CBR is now being recognized as an effective problem solving methodology, which constitutes a number of phases: case representation, indexing, similarity comparison, retrieval and adaptation. For complicated real world applications, some degree of fuzziness and uncertainty is almost always encountered. Soft computing techniques, such as fuzzy logic, neural networks and genetic algorithms will be very useful in areas where uncertainty, learning or knowledge inference are part of a system's requirements. In order to gain an understanding of these techniques, so as to identify their use in CBR, we briefly summarize their role in the following sections.

3.2.1. Using Fuzzy Logic. Fuzzy set theory has been successfully applied to computing with words [6] or the matching of linguistic terms for reasoning. In the context of CBR, when quantitative features are used to create indexes, it involves conversion of numerical features into qualitative terms for indexing and retrieval. These qualitative terms are always fuzzy. Moreover, one of the major issues in fuzzy set theory is measuring similarities, in order to design robust systems. The notion of similarity measurement in CBR is also inherently fuzzy in nature. For example, Euclidean distances between features are always used to represent the similarity among cases. However, the use of fuzzy set theory for indexing and retrieval has many advantages [7] over such crisp measurements, for example:

- Numerical features could be converted to fuzzy terms to simplify comparison.
- Fuzzy sets allow multiple indexing of a case on a single feature with different degrees of membership.
- Fuzzy sets make it easier to transfer knowledge across domains.
- Fuzzy sets allow term modifiers to be used to increase the flexibility in case retrieval.

Another application of fuzzy logic to CBR is the use of fuzzy production rules to guide case adaptations. For example, fuzzy production rules may be discovered from examining a case library and associating the similarity between problem features and solution features of cases.

3.2.2. Using Neural Networks. Artificial Neural Networks (ANNs) are usually used for learning and the generalization of knowledge and patterns. They are not appropriate for expert reasoning and their abilities for explanation are extremely weak. Therefore, many applications of ANNs in CBR systems tend to employ a loosely integrated approach where the separate ANN components have some specific objectives such as classification and pattern matching. Neural networks offer benefits when used for retrieving cases because case retrieval is essentially the matching of patterns and neural networks are very good for this task. They cope very well with incomplete data and imprecise inputs, which is of benefit in many domains, as sometimes some portion of the features is important for a new case while other features are of little relevance. Domains that use the case-based reasoning technique are usually complex. This means that the classification of cases at each level is normally non-linear and hence for each classification a multi-layer network is required.

Hybrid CBR and ANNs are a very common architecture for applications to solve complicated problems. Knowledge may first be extracted from the ANNs and represented by symbolic structures for later use by other CBR components. Alternatively, ANNs could be used for retrieval of cases where each output neuron represents one case.

*3.2.3. **Using Genetic Algorithms.*** Genetic Algorithms (GA) are robust, parallel adaptive techniques, which are used to solve search and optimization problems, inspired by the biological principles of natural selection and genetics. Learning local and global weights of case features [8] is one of the most popular applications of GA to CBR. These weights indicate how important the features within a case are with respect to the solution features. Information about these weights can improve the design of retrieval methods, and the accuracy of CBR systems.

3.3. Some CBR Tasks for Soft Computing Applications

As a summary, some of the tasks in the four REs (i.e., Retrieve, Reuse, Revise and Retain) of the CBR cycle which have relevance to be considered as prospective candidates for soft computing applications are as follows:

(1) *Retrieve*: fuzzy indexing, connectionist indexing, fuzzy clustering and classification of cases, neural fuzzy techniques for similarity assessment, genetic algorithms for learning cases similarity, probability and/or Bayesian models for case selection, case-based inference using fuzzy rules, fuzzy retrieval of cases, fuzzy feature weights learning, rough set based methods for case retrieval.

(2) *Reuse*: reusing cases by interactive and conversational fuzzy reasoning, learning reusable case knowledge, neural fuzzy approaches for case reuse.

(3) *Revise*: adaptation of cases using neural networks and evolutionary approaches, mining adaptation rules using rough set theory, learning fuzzy adaptation knowledge from cases.

(4) *Retain*: redundant cases deletion using fuzzy rules, cases' reachability and coverage determination using neural networks and rough set theory, determination of case-base competence using fuzzy integrals.

Although we have mentioned mainly the application of individual soft computing tools to the aforesaid four REs, their different combinations can also be used [9–11].

Acknowledgment

Financial support from the Hong Kong CERG research grant B-Q379 is acknowledged.

References

1. A. Aamodt and E. Plaza, "Case-based reasoning: Foundational issues, methodological variations, and system approach," *AI Communications*, vol. 7, no. 1, pp. 39–59, 1994.
2. J.L. Kolodner, *Case-Based Reasoning*, Morgan Kaufmann: San Mateo, CA, 1993, pp. 27–28.
3. I. Watson, *Applying Case-Based Reasoning: Techniques for Enterprise Systems*, Morgan Kaufmann: San Francisco, CA, 1997.
4. L.A. Zadeh, "Soft computing and fuzzy logic," *IEEE Software*, pp. 48–56, 1994.
5. Z. Pawlak, *Rough Sets, Theoretical Aspects of Reasoning About Data*, Kluwer Academic: Dordrecht, 1991.
6. S.K. Pal, L. Polkowski, and A. Skowron (Eds.), *Rough-Neural Computing: Techniques for Computing with Words*, Springer-Verlag: Berlin, 2004.
7. B.C. Jeng and T.P. Liang, "Fuzzy indexing and retrieval in case-based systems," *Expert Systems with Applications*, vol. 8, no. 1, pp. 135–142, 1995.
8. S.K. Pal and S.C.K. Shiu, *Foundations of Soft Case-Based Reasoning*, John Wiley: NY, 2004.
9. S.K. Pal and S. Mitra, *Neuro-Fuzzy Pattern Recognition: Methods in Soft Computing*, John Wiley: NY, 1999.
10. S.K. Pal and A. Skowron (Eds.), *Rough Fuzzy Hybridization: A New Trend in Decision Making*, Springer-Verlag: Singapore, 1999.
11. S.K. Pal, T.S. Dillon, and D.S. Yeung (Eds.), *Soft Computing in Case-Based Reasoning*, Springer-Verlag: London, 2001.

Simon C.K. Shiu is an Assistant Professor at the Department of Computing, Hong Kong Polytechnic University, Hong Kong. He received M.Sc. degree in Computing Science from University of Newcastle Upon Tyne, U.K. in 1985, M.Sc. degree in Business Systems Analysis and Design from City University, London in 1986 and Ph.D. degree in Computing in 1997 from Hong Kong Polytechnic University. He worked as a system analyst and project manager between 1985 and 1990 in several business organizations in Hong Kong. His current research interests include Case-base Reasoning, Machine Learning and Soft Computing. He has co-authored (with Professor Sankar K. Pal) a research monograph Foundations of Soft Case-Based Reasoning published by John Wiley in 2004.

Dr. Shiu is a member of the British Computer Society and the IEEE.

Sankar K. Pal is a Professor and Distinguished Scientist at the Indian Statistical Institute, Calcutta. He is also the Founding Head of Machine Intelligence Unit. He received the M. Tech. and Ph.D. degrees in Radio physics and Electronics in 1974 and 1979 respectively, from the University of Calcutta. In 1982 he received another Ph.D. in Electrical Engineering along with DIC from Imperial College, University of London. During 1986–87, he worked at the University of California, Berkeley and the University of Maryland, College Park, as a Fulbright Post-doctoral Visiting Fellow; and during 1990–92 & in 1994 at the NASA Johnson Space Center, Houston, Texas as an NRC Guest Investigator. He is a Distinguished Visitor of IEEE Computer Society (USA) for the Asia-Pacific Region since 1997, and held several visiting positions in Hong Kong and Australian universities during 1999–2003.

Prof. Pal is a Fellow of the IEEE, USA, Third World Academy of Sciences, Italy, International Association for Pattern Recognition, USA, and all the four National Academies for Science/Engineering in India. His research interests include Pattern Recognition and Machine Learning, Image Processing, Data Mining, Soft Computing, Neural Nets, Genetic Algorithms, Fuzzy Sets, and Rough Sets, Web Intelligence and Bioinformatics. He is a co-author of ten books and about three hundred research publications.

He has received the 1990 S. S. Bhatnagar Prize (which is the most coveted award for a scientist in India), and many prestigious awards in India and abroad including the 1993 Jawaharlal Nehru Fellowship, 1993 Vikram Sarabhai Research Award, 1993 NASA Tech Brief Award (USA), 1994 IEEE Trans. Neural Networks Outstanding Paper Award (USA), 1995 NASA Patent Application Award (USA), 1997 IETE—Ram Lal Wadhwa Gold Medal, 1998 Om Bhasin Foundation Award, 1999 G. D. Birla Award for Scientific Research, 2000 Khwarizmi International Award (1st winner) from the Islamic Republic of Iran, the 2001 INSA-Syed Husain Zaheer Medal, and the FICCI Award 2000–2001 in Engineering and Technology.

Prof. Pal has been serving (has served) as an Editor, Associate Editor and a Guest Editor of IEEE Trans. Pattern Analysis and Machine Intelligence, IEEE Trans. Neural Networks, IEEE Computer, Pattern Recognition Letters, Neurocomputing, Applied Intelligence, Information Sciences, Fuzzy Sets and Systems, Fundamenta Informaticae and Int. J. Computational Intelligence and Applications; and a Member, Executive Advisory Editorial Board, IEEE Trans. Fuzzy Systems, Int. Journal on Image and Graphics, and Int. Journal of Approximate Reasoning.