On the Automation of Case Base Development from Large Databases

D. Patterson, W. Dubitzky, S. S. Anand, J. G. Hughes
Northern Ireland Knowledge Engineering Laboratory,
University of Ulster at Jordanstown,
Newtownabbey, County Antrim,
Northern Ireland
e-mail: { wd.patterson, ss.anand w.dubitzky jg.hughes}@ulst.ac.uk

Abstract

Recent applications of Case-Based Reasoning (CBR) in industry have highlighted two major difficulties with developing CBR systems. These are, the integration of types of knowledge which are orthogonal to the knowledge represented by cases and the case engineering bottleneck. Exploiting the affinity between Data Mining and CBR we propose a unifying framework addressing these issues. Emphasising the automation of the entire CBR process we outline the Data Mining paradigms that may be employed and present initial results.

Introduction

Representing knowledge similar to that held by humans, within a computer system, has been one of the main foci in Artificial Intelligence research. Traditional rule-based approaches were found to have limitations especially in the areas of maintenance, brittleness and knowledge acquisition. CBR entered the Artificial Intelligence arena offering solutions to some key problems that had been plaguing the Rule-Based paradigm. The study of CBR is driven by two motivations: the desire to model human reasoning as pursued within cognitive science and the pragmatic desire to develop more effective and efficient computer systems. After almost ten years of both theoretical and applied experience in building and fielding case-based systems, it has been realised that the "case approach" is not free from problems. Two of the more important issues that need to be addressed are, firstly, the integration of types of knowledge which is orthogonal to the shallow and specific nature of the knowledge that is represented by cases and secondly, the automation of the case engineering process.

The main motivations behind knowledge integration lie in enhancing the epistemological adequacy (expressiveness, flexibility) of the underlying representation system, and in improving the robustness of the system's reasoning processes and components. This finding is also supported by results from cognitive science which indicate that people combine several types of knowledge when solving non-trivial problems or interpreting complex situations (Aamodt, 1991).

Three important classes of knowledge that should be considered for integration have been identified in the literature: general knowledge (shallow associational, deep conceptual), incomplete or uncertain knowledge (possibilistic, probabilistic), and contextual knowledge. Clearly, the knowledge integration issue raises the questions of how the various types of knowledge should be combined with case knowledge, and, perhaps more importantly, how the complementary knowledge is to be established or generated.

Case engineering refers to the process of *generating* and *updating* those components that represent the application-specific knowledge contained in a CBR system. The *generation* phase is concerned with determining the core knowledge structures that are needed to build a system; these include seed case identification, case content structure, indexing and case organisation structure, similarity assessment knowledge, and adaptation knowledge. Once in operational mode, cases are added to and removed from the case base (basic learning), and the ancillary knowledge structures may need to be modified in order to optimise the performance and improve the competence of the system (advanced learning). The processes that add cases, remove cases, and adjust the ancillary knowledge structures are collectively referred to as *update* operations.

Although feasible in principle, knowledge acquisition for complex applications is usually laborious and time-consuming. An important observation about these applications is that they often require substantial input from both knowledge engineers and domain specialists. For some projects, this circumstance may render the case engineering effort to be prohibitively high. If CBR is to be successful in areas where traditional case engineering methods are impractical or too expensive, the process of generating and updating case knowledge needs to be automated.

The most important motivation for this work is to *utilise databases* that already exist within many complex-application environments and apply Data Mining methods to automatically establish and maintain the knowledge components needed for CBR.

Data Mining refers to the exploration and analysis, by automatic or semiautomatic means, of large quantities of data in order to discover meaningful patterns and rules. Based on the goal to be achieved, a particular Data Mining task is undertaken using an appropriate paradigm that sifts through large data sets and arrives at appropriate knowledge. It is the authors' belief that these fundamental Data Mining tasks can provide the framework for automatically generating and maintaining CBR systems from existing, possibly large databases. Most of these tasks and their underlying algorithms are already available to the investigators in the form of the Mining Kernel System (MKS) which has been developed by the author and their co-workers (Anand *et al.*, 1997). The MKS serves as a platform to implement, test, and evaluate the automatic generation, update, and optimisation of case-knowledge from databases.

A Methodology for Data Mining and CBR Integration

Underlying Data Mining and CBR are the same set of assumptions about the world that guide the two approaches. Firstly, *regularity*, the same actions, manipulations and operations carried out in the same or similar circumstances will often lead to the same or similar outcomes or results. Secondly, *typicality*, situations, episodes, and events have the tendency to repeat. Thirdly *consistency*, small changes in the world require only small changes in our conceptions about the world, and small changes in the way we adapt to these changing circumstances. Finally, *ease of adaptation*, although situations and events rarely repeat exactly, the difference between two sets of circumstances are often small, and small differences are easy to compensate for. This affinity between the fundamental assumptions of Data Mining and CBR suggests that the two paradigms may have more to offer to each other than is currently being reflected in the respective areas. For example, the widespread perception that Data Mining is only about patterns and rules may have to be extended to include individual entities (data records) that "have taught or have the potential to teach a lesson".

In general, this work seeks to investigate the relationship between Data Mining and CBR, and how the two technologies can complement each other. Of particular interest is the study of Data Mining processes, methods, and techniques in the context of knowledge integration in CBR and automated case engineering. The principal idea is to use the data stored in an already existing database as a basis for constructing and updating case-knowledge structures such as feature saliency, case saturation, and so on. Furthermore, if more general knowledge is discovered, it should be retained and possibly integrated with the existing case-knowledge structures and processes (for example, in the form of an indexing regime that organises cases around concepts and concept relationships).

Two aspects of automatic knowledge discovery and maintenance of case-knowledge from databases can be distinguished. Firstly, the generation of the relevant knowledge structures; this relates to the initial case engineering or knowledge acquisition exercise in CBR. Secondly, the optimisation of the knowledge structures of an existing case base; this task is carried out when the system is already in operation, it is akin to the learning phase of the case-based reasoning-cycle. The emphasis in the following discussion is placed on these two processes and the corresponding components. Note, the two dimensions should not be viewed as completely uncoupled entities, but they ought to be looked at as two aspects of the same general concept.

Automatic Case Engineering

Defining the knowledge structures for a CBR system involves two orthogonal representation dimensions. One dimension is concerned with the representation of the cases themselves, their components, and the corresponding case-to-case processes such as similarity, adaptation, and repair. The other dimension deals with organising the cases within a case base such that relevant cases are retrieved when needed and that learning can take place. Clearly, the structures and processes of this dimension must work in concert with the structures and operations reflected on the level of individual cases. The following outlines how Data Mining techniques may be used to automate these processes.

Generating Case-Level Knowledge Structures from Databases

Seed case identification along with the representation and definition of the *content structure* of the cases, *feature saliency*, and *similarity* are arguably the most important issues both case engineers and subject matter specialists have to face when building a case base. We have earlier (Anand et al., 1998a) outlined how cluster analysis may be used to identify seed cases to be initially included in the case base from large databases. This coupled with a classification paradigm was used to identify discrete clusters as well as potential outliers or exceptions achieving a 1:11 reduction from database to case base size.

In general, the content of a case consists of three constituent parts, namely, *description*, *solution*, and *outcome*. There are at least three Data Mining techniques which could be used to establish the case content structure from a large data set, these are: *clustering*, *rough set analysis*, and *principle component analysis*. These processes are to discover the features that serve to define the case structure. Related to this case engineering task is the issue of determining the *saliency*, *importance*, or

simply *weight* of case features. A number of Data Mining techniques could be employed to automate this process, which include *information theoretic measures, neural networks* and *genetic algorithms*. Based on a Coronary Heart Disease case-base we showed that genetic algorithms can be successfully employed to establish *local* feature weights in such a way that the overall performance of the system is improved (Dubitzky *et al.*, 1998). Further, we have performed a comparative study of how sensitivity analysis performed on trained neural networks and genetic algorithm based search perform with respect to the identification of global feature weights (Anand et al. 1998b).

Having determined the structure of cases, the crucial issue in CBR is to represent and define similarity. Many case-based systems use some form of *distance metric* for assessing similarity. The problem with these approaches is that they do not take into account domain-specific knowledge. Using a database as a rich repository of domain-specific information, Data Mining techniques could be applied in order to generate similarity measures that are sensitive to the application area for which they are intended. In (Dubitzky *et al.*, 1997b) and (Dubitzky, 1997c) we demonstrated the use of fuzzy techniques to represent domain-specific similarity measures. In (Anand et al. 1998b) we demonstrate how statistical techniques may be used to enhance similarity metrics.

In CBR, *adaptation* is applied after a problem with the initial old solution has been pointed out or during solution formulation. At least ten different adaptation methods are reported in the literature (Leake, 1996). Again, Data Mining methods such as *rule discovery*, *classification*, *clustering*, and *genetic algorithms* constitute promising candidates for automating this case engineering procedure. One such approaches is discussed in (Hanney et al, 1996) and an alternative strategy outlined in (Anand et al., 1998a).

Generating Case Base Organisation Structures from Databases

Cases need to be organised in the case memory so as to facilitate their efficient and effective retrieval. This is commonly referred to as the *indexing* problem. Additionally, these structures and processes have to be arranged so that they can be dynamically changed enabling the system to *learn — dynamic memory* (Schank, 1982). Although orthogonal to case-level components, indexing structures and processes are usually not completely independent from the knowledge structures pertaining to cases. A rich variety of indexing schemes have been proposed in the literature ranging from "flat", vector-like schemes over so-called memory organisation packages to knowledge-intensive approaches involving sophisticated concept taxonomies (Brown, 1993). Given a large data set, Data Mining techniques could help to automate the acquisition of indexing structures in a number of ways. Firstly, knowledge-rich case base organisation strategies have been represented by means of the *semantic net* formalism (Aamodt, 1991; Brown, 1993). This organisation of a case base has the appealing feature that general and deep domain knowledge is tightly integrated with case-knowledge structures. The key issue in this approach is to reflect the concepts and their relationships. The clustering and classification methods from Data Mining provide highly suited mechanisms for generating the concepts and relationships needed to model a semantic-net-based case memory. Secondly, organising the cases in a case memory around a taxonomic indexing structure is also a powerful mechanism to tightly couple general domain knowledge with case-knowledge. Conceptual and hierarchical clustering methods are well-suited for automatically partitioning a given data set into classes and subclasses. Thirdly, the notion of a prototype case has been used in CBR to structure case bases and integrate the domain ontology into the case memory (Goel et al, 1990). In Data Mining, prototypes arise as a result of classification and clustering tasks. Thus, both classification and clustering could contribute to the automatic determination of case indexing knowledge.

Updating Case-Knowledge Structures from Databases

The process of automatically updating a case base is concerned with the optimisation and maintenance of the case base's knowledge structures. This process is different to the case-knowledge generation phase in that the necessary knowledge structures are already in place but require revision in the light of problem-solving experience with new problems. In CBR parlance the update process is equivalent to the *learning cycle*. However, since the update mechanisms discussed here are also driven by the changes made to the associated operational database, they go beyond the traditional learning procedures. It is precisely this database back-end that makes it possible to explore a variety of automated learning processes so far not conceived in conventional CBR systems.

Perhaps one of the more significant breakthroughs that could be achieved by using large databases as support repository for CBR is a solution to the so-called *swamping problem* (Smyth & Keane, 1995). The swamping problem dictates that only those cases which are necessary to maintain a certain competence level of the entire system should be retained in the case library. Clearly, a number of Data Mining techniques could be employed to automatically determine cases (data records) in a database that are likely to contribute more to the competence of the case base than others. Typical candidates of such cases are those that arise from the application of clustering and

classification methods, that is, prototype cases (norms) and outliers (exceptions) (Anand et al., 1998a).

Genetic programming is rapidly developing into a promising technique for tackling complex optimisation tasks where *hillclimbing*, *simulated annealing*, and conventional genetic algorithm methods fail (Koza, 1992). In contrast to genetic algorithms, genetic programming techniques, allow domain-specific knowledge to be incorporated into the algorithm. In most case-based systems, the cases tend to be highly structured entities (Aamodt 1991). This suggests that, given an operational database as a back-end, genetic programming could be used to optimise a case base with respect to both the cases held in the case memory and the saliency of the case features. In addition to addressing the swamping problem, the changes occurring in the associated operational database could be exploited to *update* the case-base with respect to its ancillary knowledge structures such as adaptation, similarity, and indexing knowledge. The update of these components is intended to increase the competence of the CBR system.

The use of competence feedback for updating retrieval mechanism through the definition of exception spaces and KINS has been shown to be viable as well as desirable by the authors' (Anand et al., 1998c).

Conclusion

In this paper Data Mining techniques were proposed as a methodology for automating the entire CBR process. Each proposed step in this process was outlined and the particular Data Mining technique applicable discussed. The advantages of this automation in terms of knowledge acquisition, maintenance and case-base construction were shown and the results of initial work carried out by the authors given. Overall the results are very encouraging confirming the validity of the outlined approach.

References

Aamodt, A. 1991. A Knowledge-Intensive Approach to Problem Solving and Sustained Learning, PhD dissertation, University of Trondheim, Norwegian Institute of Technology.

Anand, S. S.; Scotney, B. W.; Tan, M. G.; McClean, S. I.; Bell, D. A.; Hughes, J. G.; and Magill, I. C. 1997. Designing a Kernel for Data Mining. *IEEE Expert* 12(2): 65 - 74.

Anand, S. S.; Patterson, D.; Hughes, J. G.; and Bell, D. A. 1998a. Discovering Case Knowledge using Data Mining. In *Proceedings of the Pacific-Asia Conference in Knowledge Discovery and Data Mining*, Springer-Verlag,

Anand, S. S.; and Hughes, J. G. 1998b. Hybrid Data Mining Systems: The Next Generation. In *Proceedings. of the Pacific-Asia Conference in Knowledge Discovery and Data Mining*, Springer-Verlag.

Anand, S. S.; Patterson, D. W.; Hughes, J. G. 1998c. Knowledge Intensive Exception Spaces, To appear in Proceedings of AAAI-98.

Brown, M. 1993. A Memory Model for Case Retrieval by Activation Passing", PhD Dissertation, Department of Computer Science, University of Manchester, Manchester, UK.

Dubitzky, W.; Hughes, J. G.; Bell, D. A.. 1998. Discovery of Local Feature Weights to Improve Retrieval in Case-Based Reasoning: A Genetic Algorithm Approach, Proc. KDD, 1998. (submitted)

Dubitzky, W.; Schuster, A.; Bell, D. A.; Hughes, J. G.; Adamson, K. 1997. How Similar is VERY YOUNG to 43 Years of Age? On the Representation and Comparison of Polymorphic Properties, in *Proc. 15th International Joint Conference on Artificial Intelligence*, pp226-231, Japan,

Dubitzky, W. 1998. Knowledge Integration in Case-Based Reasoning: A Concept-Centred Approach, PhD thesis, University of Ulster, School of Information and Software Engineering.

Goel, A.; Chandrasekaran, B. 1990. A Task Structure for Case-Based Design, in *Proc. 1990 IEEE International Conference on Systems, Man, and Cybernetics*, pp587-592, IEEE Systems, Man, and Cybernetics Society Press.

Hanney, W.; Keane, M. T. 1996. Learning Adaptation Rules from a Case-Base, in *Proc. Advances in Case-Based Reasoning, 3rd European Workshop, EWCBR-96*, pp179-192, Lausanne, Switzerland.

Koza, J. R. 1992. *Genetic Programming: On the Programming of Computers by means of Natural Selection*, The MIT Press.

Leake, D. B. (editor). 1996. *Case-Based Reasoning: Experiences, Lessons & Future Directions*, MIT Press, MA.

Schank, R. C. 1982. *Dynamic Memory: A Theory of Learning in Computers and People*, Cambridge University Press.

Smyth, B.; Keane, M.T. 1995. Remembering to Forget: A Competence-Preserving Case Deletion Policy for Case-Based Reasoning Systems, in *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pp337-382.

Appendix

1. Integration name/category: CBR/Data Mining
2. Performance Task: Any (Depending on Database)

3. Integration Objective: Efficiency in time taken to build CBR system. Data mining assists in discovery of initial seed cases from Database, Indexing, discovering case structure & adaptation knowledge.
4. Reasoning Components: Rule Induction & self organising maps for discovery of seed cases & their indexing. Genetic Algorithms for case structure optimisation. Rule Induction for competence feedback.
5. Control Architecture: Unified with CBR as end product of Data Mining. Data Mining also used as competence feedback.
6. CBR Cycle Step(s) Supported: *Pre-processing* - Identification of seed cases using self organising maps.

   Retrieval - high level indexing using Rule Induction & discovery of case structure using Genetic Algorithms

   Reuse -Discovery of adaptation knowledge using Rule Induction

   Revision - Competence feedback through discovery of KINS using Rule Induction & Genetic Algorithms

   Retention - implicit within revision methodology

   Post processing - Revision phase utilising competence feedback.
7. Representations: Cases & rules for competence feedback & adaptation.
8. Additional Components: Interface to Database
9. Integration Status: Proposed with initial empirical evaluation of individual components.
10. Priority future work: Integration of individual components & empirical evaluation & application.