Towards Case Completion with inferencing and solution identification using 'Nested CBR'

Kent Ramchand
Swinburne University of Technology, kramchand@swin.edu.au

Mohan Baruwal Chhetri
Swinburne University of Technology, mchhetri@swin.edu.au

Ryszard Kowalczyk
Swinburne University of Technology, rkowalczyk@swin.edu.au

Towards Case Completion with inferencing and solution identification using 'Nested CBR'

Research-in-progress

Kent Ramchand
School of Computer Science and Software Engineering,
Swinburne University of Technology, Melbourne, Australia
Email: kramchand@swin.edu.au

Mohan Baruwal Chhetri
CSIRO Data61, Melbourne, Australia
mohan.baruwalchhetri@csiro.au
School of Computer Science and Software Engineering,
Swinburne University of Technology, Melbourne, Australia
Email: mchhetri@swin.edu.au

Ryszard Kowalczyk
School of Computer Science and Software Engineering,
Swinburne University of Technology, Melbourne, Australia
Email: rkowalczyk@swin.edu.au
Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland
Email: rkowalcz@ibspan.waw.pl

Abstract

Case Based Reasoning (CBR) provides a framework to capture past problems and their solutions to solve future problems. Problem cases are typically complete; however, it is not always possible to have a complete problem case due to complexity, lack of data, or availability of human expertise. The limitations of existing approaches for handling incomplete cases include a reliance upon manual input, such as Conversational CBR (CCBR) and Incremental CBR (ICBR), or a rigid structure of relationships maintained using a semantic ontology, to infer the missing feature values. Using the case base to infer feature values increases the efficiency and likelihood of identifying a relevant solution compared with manual interactions because the case base is based upon proven problem to solution correlation. Therefore, in this work-in-progress paper, we propose 'Nested CBR' as an approach for the automated completion of partial problem cases, and the subsequent solution identification, thereby avoiding manual input and improving solution efficiency and meaning.

Keywords Case-based Reasoning, Partial cases, Inference techniques and Similarity.

1 Introduction

Case Based Reasoning (CBR) (Watson, 1999) is an approach for solving new problems, typically represented using features, through learning from the solutions to past similar problems. Determining the most relevant prior cases is a fundamental problem in CBR that requires more elaborate techniques when a full problem description is not available, i.e., there are missing but still required feature values. Each feature in a problem case has a corresponding value and priority. The priority indicates its importance/relevance to the problem, relative to the other features. Typically, the more feature values that are known, the easier it is to identify the most relevant cases from the case base and therefore a suitable solution. However, it is not always possible to have a complete problem case to begin with, and there can be several reasons that can contribute to missing feature values including oversight, problem complexity, lack of data, or availability of human expertise. As an illustrative example, correct decision-making about the cloud platform architecture is crucial for the success of any cloud migration project because wrong decisions can lead to unintended consequences including project delays, budget overruns, application instability, and technical debt (Ramchand et al., 2018). However, the responsibility of decision-making is increasingly moving into the hands of the business sponsors who do not have a full set of requirements to begin with, i.e., they must start with a partial problem description.

Currently, there are four main approaches for deriving feature values for partial cases, (a) ignore the missing data and continue with CBR (Van Buuren, 2012), (b) use Conversational CBR (CCBR) (Jurisica et al., 2000) or Incremental CBR (ICBR) (Jurisica et al., 2000), (c) apply simple arbitrary rules to calculate mean, minimum or maximum feature values (Bogaerts, Leake, 2004) from the case base, or (d) leverage ontology for a specific domain (Qin et al., 2018). The shortcoming with using arbitrary rules to identify potential feature values is that the associated assumptions are not being tested and are therefore unlikely to be targeted at identifying the most relevant solution. Alternatively, using ontologies requires the combinations of feature values and their relationships to be well-understood and modelled to then identify meaningful missing feature values, which leads to a rigid model. ICBR and CCBR (Jurisica et al., 2000) require manual input from the user who may not have the information available, nor the time and expense to invest and respond. CCBR is an extension of foundational CBR that takes the approach of introducing questions and receiving answers from the CBR user to provide feature values when there are missing feature values. ICBR incorporates CCBR through querying the case base with the feature values provided, then using CCBR to gain input from the user manually, then performing a query of the subset of cases retrieved initially to minimise computation.

In this paper, we propose 'Nested CBR' as an approach for case completion and solution identification that uses a two-phase strategy with inference techniques to (a) filter cases from the case base in the first instance to identify the most meaningful cases, and, (b) use a data driven similarity metric for proportionally inferring the remaining feature values to complete the problem case. The motivation for our approach is for problem solving efficiency by re-using information from the case base versus ignoring the missing values or using simplistic rules. In the situation where the user does not nominate any feature values, our approach recommends feature values using CBR principles from the case base. Having been completed, the completed cases are stored in the case base and become available for use in 'Nested CBR'.

There are four key benefits of using 'Nested CBR' with inferencing techniques for the identification of missing case feature values: (a) it derives feature values proportionally from each shortlisted case, based on the degree of similarity, instead of relying on implied assumptions or arbitrary rules, (b) having identified the missing feature values, it then proposes a solution to the complete problem with inferred values, (c) the completed cases with inferred values can be stored in the case base and used in 'Nested CBR' for new partial cases; however, they require a weighting to measure their degree of completeness, and (d) the process of deriving feature values in 'Nested CBR' is automated. Essentially, the learning in the case base is exploited to first recommend feature values and then a solution based upon the completed case.

The novel contribution of our research is to provide a 'Nested CBR' framework for the automated generation of solutions for new problems. CBR is first used for the completion of partial problem cases, using inferencing techniques that are more effective and meaningful than existing techniques that use arbitrary or simplistic rules, or require complex relationships between features and values per problem domain that must be maintained. CBR is again used to identify a relevant solution for the completed problem case. To the best of our knowledge, our proposed 'Nested CBR' framework is the first approach to use CBR to derive missing feature values to complete a problem case and then iterate through the CBR phases again to identify a solution for the complete case with inferred values.

The rest of the paper is organised as follows. Section 2 presents related work on strategies to complete partial cases in CBR. It then compares related research for case completion and proposes a new approach called 'Nested CBR'. Section 3 presents an overview of the 'Nested CBR' framework including the use of similarity and inferencing techniques to derive feature values to complete the case using CBR principles and then to identify a relevant solution also using CBR principles. It is followed by an illustrative example in Section 4. Section 5 concludes the paper by providing a summary of the completed work and identifying areas of future work.

2 Related Research

This section provides a review of existing CBR approaches and methods relevant to this research. There are multiple reasons that can contribute to missing feature values resulting in incomplete problem cases including oversight, costs related to obtaining Subject Matter Expert (SME) input, lack of time or experimentation by the CBR application user (Van Buuren, 2012). Our approach for the development of recommendations is to propose inference techniques based upon data driven similarity (Jaiswal et al., 2019) with traditional CBR steps for partial cases and CBR again to propose a solution to the completed case with inferred values. The missing feature values are calculated after the Case Discovery and Retrieval step in the first iteration of CBR using inferencing techniques.

There are four main existing approaches for deriving feature values for partial cases in research, (a) ignore the missing data and continue with CBR (Van Buuren, 2012), (b) use Conversational CBR (CCBR) (Jurisica et al., 2000) or Incremental CBR (ICBR) (Jurisica et al., 2000), (c) calculate mean, minimum or maximum feature values (Bogaerts, Leake, 2004) from the case base, or (d) leverage an ontology for a specific domain (Qin et al., 2018). Our proposed automated inferencing technique leverages the case base to recommend case values proportionally from the most relevant cases versus requiring manual input from the CBR system user using a technique like CCBR.

When using ICBR incorporating CCBR to complete a case having missing feature values, the system interacts with the CBR user to elicit feature values for those features that are 'free' or easy to come by. 'Free' feature values are populated in the problem case and used to query the case base to then identify the 'Expensive' feature values through further interaction with the user. Each future search of the case base uses a refined set of cases using the resulting subset of the previous query for further refinement (Smyth et al., 1998) to find the harder to identify feature values. The shortcoming of this approach is that requirements elaboration is largely manual and relies upon the end user being able to discern the harder to identify feature values from the remaining set of cases.

The authors Bogaerts and Leake (2004) proposed four different strategies for assessing similarity in partially described cases. The Default Distance strategy assigns a fixed distance measure whenever the problem case has missing feature values that can lead to misleading solutions being recommended. Alternatively, the Full Mean strategy exploits global feature information and replaces missing feature values with the mean values for those features from all the cases in the case base. The Nearest Neighbour Mean (NN Mean) strategy uses local information, leveraging the mean values of nearby cases to the supplied feature values in the partial case to recommend feature values to complete the case. Although the most meaningful cases are identified, the limitation with this approach is that it uses the mean values from the nearest cases to recommend the missing feature values. The Region Mean strategy tries to combine the benefits of the Full Mean and NN Mean strategies by pre-computing the means for clusters in the case base and then predicting means based on the nearest precomputed cases to the problem case. The conclusion drawn by the authors from their experiments were that simplistic methods such as 'Near By' and 'Full Mean' resulted in poor solution recommendations. 'Nested CBR' is similar to the NN Mean technique, however, to improve the likelihood of 'fit for purpose' solutions, it uses more sophisticated strategies for inference of missing feature values.

An alternative method of completing partial cases is to use an index driven technique (Wiratunga et al., 2003) that identifies the cluster of cases that map to the partial problem case to guide case selection to elicit the missing case data. The completeness of the case history is critical for the recommendations to be meaningful. Alternatively, CBR can be combined with the Constraint Satisfaction Problem (CSP) technique (Marling, 2002) that uses information in the candidate case and the case base to find the missing feature values subject to restrictions on which combinations of static features and their values and business rules are applicable. A shortcoming of this approach is the complete problem statement may not be identified because fixed business rules and value ranges are used that limits the data being used to identify missing feature data. For example, unexpected business rules or values in a number range may emerge that invalidate the approach or make the model complex to maintain. An example of a domain specific approach where CBR is used in conjunction with an ontology (Mabkhot et al., 2019) is in the field of Manufacturing. As the problem space evolves it is likely the business rules, number ranges and feature relationships will require maintenance. The number of combinations of features and values to be maintained can be considerable that directly impacts the cost of storage and labour required.

3 Nested CBR for Partial Cases

3.1 Generalised Approach to Complete Cases

The Nested CBR framework is proposed to assist a user who does not have access to the necessary information to complete a problem statement. To provide recommended feature values and a relevant solution, the Nested CBR framework uses CBR in two iterations (as shown in Figure 1).

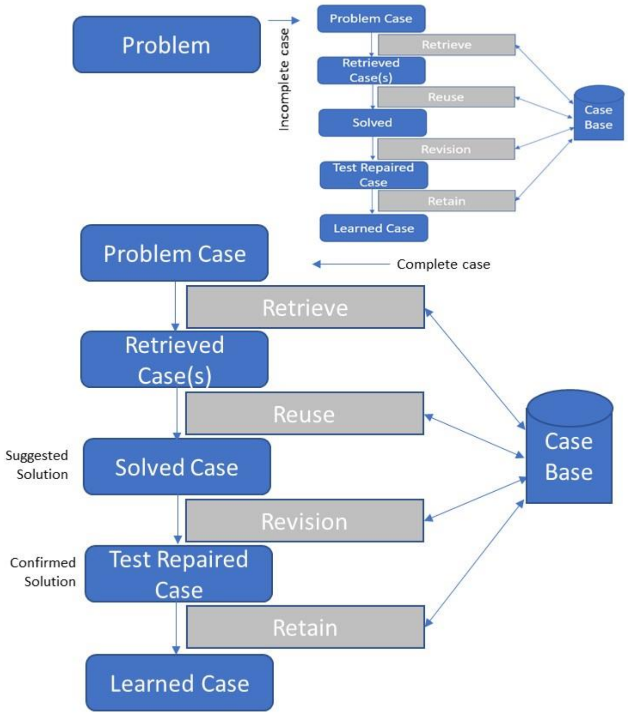

Figure 1 – Nested Case-based Reasoning

In iteration one, CBR is used to complete the partial problem case by leveraging intelligence stored in the problem case with proven problem to solution correlation. A partial problem case is presented to the CBR system and relevant cases are 'Retrieved' from the case base. Each relevant candidate case's distance is measured against the partial case for similarity, with each of them ranked according to their similarity metric. Those cases that are within the similarity threshold are used to derive the missing feature values proportionally according to the similarity metric. Having 'Solved' for the missing feature values, the partial problem case is revised to be complete.

In iteration two, a complete case (with inferred values) is presented to the CBR system to then retrieve and present the most relevant solution. The solution is reviewed and then confirmed so that the complete case is 'Solved' enabling it to be 'Repaired' and 'Retained' going forward if it is not a duplicate.

3.2 Shortlisting Cases for Partial Case Handling

Using Nested CBR for feature value derivation with inferencing techniques relies upon determining the most relevant cases to derive feature values. To identify the most relevant cases, each partially specified problem case is measured against each case in the case base to calculate a similarity metric, for example, Euclidean distance. Having been ranked using the similarity metric, the most relevant cases are shortlisted based on the CBR system user's threshold for the degree of similarity or context required. For exceptionally large databases, a meaningful subset is identified through querying the case base using the values provided to obtain a subset first, to then measure against. The similarity metric for each case plays an important role in the two-phase strategy for our inferencing technique.

3.3 Inferencing technique

The inferencing technique involves using the cases in the case base that satisfy the user's similarity requirement and then calculating each feature value proportion. Essentially, the weighted average is used to determine the proportion of each feature value to be summed in the determination of the recommended feature value. Once the weighted average has been determined for each case, the next step is to calculate the proportion of each feature value and then add them together to form the recommended feature value.

The inference technique calculates the relative degree of similarity of each of the shortlisted cases as a basis to recommend meaningful values. The case inference approach is applied as follows: (a) where no feature values are supplied, the average feature values are derived from the case base, (b) where there is one shortlisted case, its feature values are used to derive the relevant missing feature values, and (c), where at least one feature value is supplied, feature values are calculated proportionally from the shortlisted cases to recommend missing feature values to complete the case.

3.4 Second Iteration of CBR for Solution Recommendation

The second iteration of CBR is used to select one or more "most similar" cases to the completed case with inferred values for solving the newly formed case from the first iteration; the main steps include (a) retrieve cases based on the completed case with inferred values, (b) identify the most similar case by comparing each of the retrieved cases with the completed case, (c) recommend a solution based on the highest degree of similarity, and (d) Store the new case into the case base so that is then available for comparison to a new partial case.

3.5 Managing the Case Base

With the introduction of Nested CBR a new inferred completed case type will exist. Our use of CBR with a similarity assessment is determined through the strength of matching of the candidate case with fully specified or inferred problem cases in the case history. Partially specified cases, with inferred values, are those with the status: Retain-Inferred. The new status of the cases in the Case Base that manage a case lifecycle are:

Case Status	Definition
Retain-RealWorld	A Real-World fully specified case
Retain-Inferred	A Completed Case with inferred values. Note, a Retain-Inferred become a Retain-RealWorld if implemented.

Note, the inferred cases can be used for case completion in future by considering the completion factor.

4 Sample CBR for Completing a Problem Case (with Illustrative Example)

4.1 Nested CBR Approach and Model

The proposed approach for Nested CBR uses Euclidean distance as an example to measure the similarity of the problem case with shortlisted cases. This is followed by a formula to calculate the relative proportions of each feature value to form recommended feature values to progress from a partial problem case to a complete problem case. The features and priorities supplied by the user for the problem case p are normalised values between 0 and 1. The incomplete problem case p is completed as follows:

- Find similar cases based on the distance between the problem case p and the known cases $c_i, i = 1, \dots, N$, considering known feature values f_p^j only, which for example using an Euclidean distance is as follows:

$$d_i(p, c_i) = \sqrt{\sum_{j=1}^{N} pr_i^j \, (f_p^j - f_i^j)^2}, \; i = 1, \dots, N$$

- Shortlist the most similar cases that are close to the problem case, which for example do not exceed a given distance threshold D_r as follows:

$$d_i(p, c_i) \le D_r, i = 1, \dots, N$$

- Infer the missing feature values in the problem case based on shortlisted known complete cases, which for example according to their relative distance is as follows:

$$f_c^j = \frac{\sum_{\substack{i=1 \\ d_i(p,c_i)\le D_r}}^{N} (1-d_i)f_i^j}{\sum_{i=1}^{N} d_i}, j = 1, \dots, M$$

where:

- $p = [f_p^j;\ pr_p^j]$ is the problem case consisting of features $f_p^j, j = 1, \dots, M$ with the corresponding priorities $pr_p^j, j = 1, \dots, M$. It also includes the features with missing (to be completed) values f_j^c
- $c_i = [f_i^j;\ pr_i^j]$ is the i^{th} case in the case base, consisting of features $f_i^j, i = 1, \dots, N, j = 1, \dots, M$ with the corresponding priorities $pr_i^j, i = 1, \dots, N,\ j = 1, \dots, M$
- D_r is a distance threshold provided by the user

4.2 Illustrative Example of 'Nested CBR' to Infer Missing Feature Values

Based on the formulas in Section 4.1, we elaborate an example to illustrate how to progress from a partial case to a complete case. Let us consider a problem case for which three feature values and priorities are known as follows: p = [1,0.6;0.5,0.5;0.2,0.4;–,–;–,–]. Suppose that the case base contains two complete cases, c_1, [0.9;0.4;0.1;0.4;0.3] and c_2, [0.5;0.4;0.2;0.3;0.2]. The Euclidean distance between the problem case p and the two cases c_1 and c_2 is calculated as follows:

$d(p,c_1) = \sqrt{}\ (0.6(1 - 0.9)^2 + 0.5(0.5 - 0.4)^2 + 0.4(0.1 - 0.2)^2) = 0 = \sqrt{0.015} = 0.12$

$d(p,c_2) = \sqrt{}\ (0.6(1 - 0.5)^2 + 0.5(0.5 - 0.4)^2 + 0.4(0.2 - 0.2)^2) = \sqrt{}\ 0.155 = 0.39$

Thus, c_1 is closer than c_2. To calculate the inferred values, the first step is to calculate the weight of each candidate case using their respective similarity metrics:

- c_1: 1 – (0.12/0.51) = 0.76
- c_2: 1 – (0.39/0.51) = 0.24

The second step is to calculate the relative proportion of the values from each case for the respective feature values by calculating the weighted average. These values are recommended to be in the problem case:

- f_4 = (0.76 x 0.4) + (0.24 x 0.3) = 0.38
- f_5 = (0.76 x 0.3) + (0.24 x 0.2) = 0.28

These feature values are used to complete the problem case, f_p = [1.0,0.5,0.2,0.38,0.28].

Consider an extended example to calculate feature values incorporating a completed case with inferred values in the shortlisted cases that includes an extension of the problem domain for inferred values identified by if4 and if5 in p = [f_1;f_2;f_3;if_4;if_5]. The additional shortlisted completed case with inferred values, c_3, is [1;0.6;0.5;0.4;0.3]. The Euclidean distance is first calculated as follows:

$d(p,c_3) = \sqrt{}\ (0.6(1 - 1)^2 + 0.5(0.5 - 0.6)^2 + 0.4(0.2 - 0.4)^2) = \sqrt{}(0 + 0.005 + 0.02) = 0.15$

Next, the relative weight for each case is calculated as follows:

- c_1: 1- (0.12/0.66) = 0.82
- c_2: 1- (0.39/0.66) = 0.41
- c_3: 1- (0.15/0.66) = 0.77

An additional weight is derived for the completed case with inferred values identifying the number of features specified through a normalised value between 0 and 1. The weight is combined with the formula to calculate the inferred values, and is known as 'Completion factor', a normalised value between [0 …… 1]. The completion factor is calculated as follows:

$$cf = \frac{\sum f_i}{(\sum if_i + \sum f_i)}, i = 1 \ldots . I$$

The completion factor is applied to Case 3's similarity and values (3 of 5 features were specified by a user), cf = 3/5 = 0.6. Using the weight of each case, the relative proportion of the values from each of the shortlisted cases is calculated based on their ranking and therefore distance from the problem case, i.e., the closest case has the highest proportion, second closest case has the next highest, third case has the lowest proportion:

- f_4 = (0.41 x 0.4) + (0.77 x 0.3) + 0.6(0.82*0.4) = 0.59
- f_5 = (0.41 x 0.3) + (0.77 x 0.2) + 0.6(0.82*0.3) = 0.42

These feature values are used to complete the problem case, f_p = [1.0,0.5,0.2,0.59,0.42]. The role of the 'Completion Factor' in deriving missing feature values will be refined in future work.

5 Conclusion

In this paper, we have presented our work-in-progress for developing a Nested CBR framework that allows one to go from a partial problem case to a solved case. Our framework uses the first iteration of CBR, to identify the missing feature values from shortlisted similar cases, and then uses CBR again to identify a relevant solution for the completed case. Our proposed automated inferencing techniques leverage the case base to recommend feature values proportionally from the most relevant cases as compared with existing approaches such as calculating: the minimum, maximum or average feature value from the case base or requiring manual input from the CBR system user. Our ongoing research focus is to:

- Provide an approach for using both completely specified cases and cases with inferred values to derive feature values for partial cases based on a completion factor weighting.
- Introduce a new case base management process and requirement to cater for the use of completely specified cases and complete cases with inferred values.
- Evaluate the Nested CBR framework with existing inference approaches.

6 References

1. Aamodt, A., Plaza, E., 1994, "Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches", AI Communications, vol. 7, no. 1, pp. 39-59
2. Bogaerts, S., Leake, D., 2004, "Facilitating CBR for Incompletely-Described Cases: Distance Metrics for Partial Problem Descriptions", Springer Berlin Heidelberg
3. Jaiswal, A., Bach, K., 2019, "A Data-Driven Approach for Determining Weights in Global Similarity Functions". Lecture Notes in Computer Science (LNCS), Springer
4. Jurisica, Igor., Glasgow, J., Mylopoulos, John., 2000, "Incremental Iterative Retrieval and Browsing for Efficient Conversational CBR Systems", Appl. Intell.. 12. Pp 251-268.
5. Lopez De, M., McSherry, D, Bridge, D, Leake, D., 2005, "Retrieval, reuse, revision and retention in case-based reasoning". The Knowledge Engineering Review, Volume 20, Issue 3
6. Makbot, MM., Al-Samhan, MA., Hidri, L., 2019, "An Ontology-Enabled Case-Based Reasoning Decision Support System for Manufacturing Process Selection", Volume 2019, Hindawi
7. Marling, C., Sqalli, M., Rissland, E., Munoz-Avila, H., Aha, D. 2002. "Case-Based Reasoning Integrations", AI Magazine, 23(1), 69. https://doi.org/10.1609/aimag.v23i1.1610
8. Qin, Y., Lu, W., Qi, Q., Liu, L, Huang, M., Scott, P., Jiang, X., 2018, "Towards an ontology-supported case-based reasoning approach for computer-aided tolerance specification", Knowledge-Based Systems, Volume 141, pp129-147
9. Ramchand, K, Baruwal Chhetri, M. Kowalczyk, R., 2018, "Towards A Comprehensive Cloud Decision Framework with Financial Viability Assessment," In PACIS, p. 32. 2018.
10. Smyth B., Cunningham P., 1998, "Advances in Case-Based Reasoning.", Lecture Notes in Computer Science, vol 1488. Springer, Berlin, Heidelberg
11. Tamoor M., Gul H., Qaiser H., Ali, A. 2015, "An optimal formulation of feature weight allocation for CBR using machine learning techniques," SAI Intelligent Systems Conference (IntelliSys), London, pp.61-67
12. Van Buuren, S., 2012, "Flexible Imputation of Missing Data", Chapman & Hall/CRC Interdisciplinary Statistics Series, CRC Press, Baco Raton, Florida

13. Watson I, 1998, "Is CBR a technology or a methodology", Tasks and Methods in Applied Artificial Intelligence. IEA/AIE 1998. Lecture Notes in Computer Science (Lecture Notes in Artificial Intelligence), vol 1416. Springer, Berlin, Heidelberg
14. Wiratunga N., Craw S., Massie S. 2003. "Index Driven Selective Sampling for CBR", Case-Based Reasoning Research and Development

Copyright