Maintenance of Case Bases: Current Algorithms after Fifty Years

Jose M. Juarez[1], **Susan Craw**[2], **J. Ricardo Lopez-Delgado**[1], **Manuel Campos**[1]
[1] University of Murcia, Murcia, Spain
[2] Robert Gordon University, Aberdeen, UK
jmjuarez@um.es, s.craw@rgu.ac.uk, jricardo@um.es, manuelcampos@um.es

Abstract

Case-Based Reasoning (CBR) learns new knowledge from data and so can cope with changing environments. CBR is very different from model-based systems since it can learn incrementally as new data is available, storing new cases in its case-base. This means that it can benefit from readily available new data, but also case-base maintenance (CBM) is essential to manage the cases, deleting and compacting the case-base. In the 50th anniversary of CNN (considered the first CBM algorithm), new CBM methods are proposed to deal with the new requirements of Big Data scenarios. In this paper, we present an accessible historic perspective of CBM and we classify and analyse the most recent approaches to deal with these requirements.

1 Introduction

Case-based reasoning (CBR) solves new problems by retrieving similar, previously solved problems (cases) and reusing their solutions. The case-base is an essential component of any CBR system, storing a set of cases to be retrieved [Craw, 2017]. In the era of Big Data, automated maintenance of case-bases plays a critical role to guarantee the correctness and efficacy of Case-Based Reasoning (CBR) systems. According to [Goel and Diaz-Agudo, 2017], case acquisition from raw data is one of the eight challenges in CBR research. The design of novel algorithms has recently received wide attention from the community, bridging the gap between current AI techniques and the requirements of Big Data: *volume* and *velocity* of data, *variety* of data sources and *value* of the solution. Large-scale case-bases are necessary to keep the competence of the CBR engine and Case-base Maintenance (CBM) algorithms focus on revising the knowledge: deleting out-of-date cases, indexing and compacting the existing information [Leake and Schack, 2015].

Many CBM algorithms have been present in CBR systems during the last three decades. The classical approach assumes that CBM is an offline process, dealing with a finite, limited number of stored cases where the user actively queries the system with a new case to be solved [Salamó and Golobardes, 2001; Smiti and Elouedi, 2014]. Today, unprecedented flows of data demand answers continuously in a dynamic context, such as home monitoring alert systems [Lupiani *et al.*, 2017], industrial robot supervision [Chebel-Morello *et al.*, 2015] or the semantic analysis of the Wikipedia corpus [Mathew and Chakraborti, 2017]. To address such problems, a number of promising CBM algorithms have been proposed.

The contributions of this survey paper are: (1) A brief but exhaustive map of CBM algorithms: structuring the knowledge and linking past and present approaches (§2); and (2) putting 10 recent CBM algorithms in perspective: categorizing their contribution according to methodology, computational approach and comparing results through examples (§3).

2 General Perspective

This section presents a general survey of CBM. We first introduce a unified notation to ease a comparative analysis. A graphical review of CBM algorithms is shown in Fig.1, depicting the temporal evolution and dependencies using an arc diagram. Finally, we discuss major principles followed by CBM researchers and their interaction with Machine Learning techniques. We select historic algorithms to illustrate this.

2.1 Notation

Cases are the atomic unit of reasoning, describing a specific problem and its solution. Traditionally, the problem is a vector of attributes and the solution is a quantitative or qualitative value. In formal notation, the problem and solution spaces are expressed as $\Pi = A_1 \times \ldots \times A_n$ and $\Omega = \{\omega \mid \exists \pi \in \Pi : \omega \ solves \ \pi\}$ respectively. A case is the pair $c = (\pi, \omega)$ and a set of cases forms the case-base $C \in \mathcal{P}(A_1 \times \ldots \times A_n \times \Omega)$. CBR is an analogy-based model and the notions of similarity and distance are essential to compute the mechanisms of reasoning. Function $Sim : C \times C \rightarrow [0, 1]$ expresses the similarity between two cases, often calculated as the inverse of the distance function between the cases' problems: $\delta : \Pi \times \Pi \rightarrow \mathbb{R}^+$.

2.2 Nearest Neighbour Model

Case-base maintenance is understood by many researchers as a problem of reducing the number of cases and proposing algorithms to identify redundant and noisy cases. Noise reduction methods seek to increase classification accuracy while redundancy reduction's goal is to improve retrieval efficiency [Cummins and Bridge, 2011].

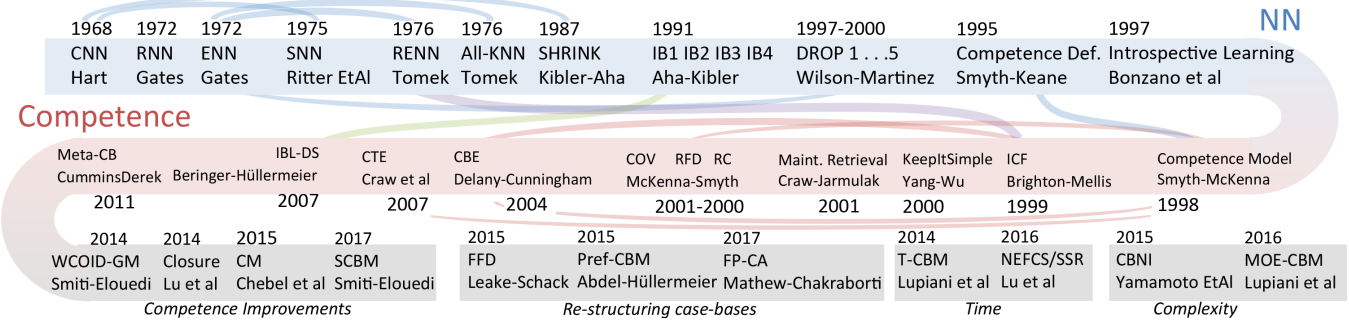

Figure 1: 50 Years CBM map: Arc diagram of selected CBM methods and principles (1968-2018)[1].

In the early decades, efforts focus on the study of nearest neighbour (NN) and instance-based learning methods adopted from Machine Learning [Salamó and Golobardes, 2001]. Some CBM examples of noise reduction are Condensed NN (CNN) considered the first algorithm for CBM [Hart, 1968], Reduced NN (RNN), Edited NN (ENN) or Selective NN (SNN) that essentially look for subsets of cases ensuring they can classify correctly former cases using different heuristics. These are known as the NN-family. The main drawback is the minimal set is not guaranteed and they are sensitive to noisy cases. To solve these problems, other algorithms were proposed, incrementing the computational complexity by iteratively run NN approaches, like All-KNN or Repeated ENN (RENN) [Tomek, 1976]. We choose RENN (Algorithm 1), to illustrate the essential mechanisms adopted by NN-family approaches following a *decremental* process. Function $correctClassifyNN(c, C, \delta, 3)$ is true when the 3 nearest neighbours of c in C, using distance function δ, have the same solution as c.

Algorithm 1 RENN: Repeated ENN (Tomek 1976)

Input original case-base C
Output maintained case-base C'
  $C' \leftarrow C$
  repeat
    % body loop is ENN algorithm (Wilson 1972)
    for all $c \in C'$ **do**
      if $NOT\ correctClassifyNN(c, C, \delta, 3)$ **then**
        $C' \leftarrow C' - \{c\}$ *% c noisy case deleted: decrement*
      end if
    end for
  until C' *no changes*
  return C'

2.3 Competence Models

Unlike the Machine Learning perspective, understanding CBM as an instance selection problem, other authors highlight the purpose of each single case in the CBR cycle. In particular, Smyth-Keane-McKenna's Competence Model had a deep impact on the field, stating the properties to determine which cases to include, according to their capacity to solve problems in the context of a CBR system [Smyth and Keane, 1995; Smyth and McKenna, 2001]. This competence is determined by two basic properties: coverage and reachability. Eq.(1)-(3) formalise the original concepts in terms of NN, called Coverage Set (CS), Reachability Set (RS) and Related Set:

$$CS(c, C) = \{c' \in C \mid c' \in NN(c, C) \wedge c\ solves\ c'\}, \quad (1)$$
$$RS(c, C) = \{c' \in C \mid c \in NN(c', C) \wedge c'\ solves\ c\}, \quad (2)$$
$$RelatedSet(c, C) = CS(c, C) \cup RS(c, C) \quad (3)$$

where $solves$ indicates that c and c' have the same solution. In other words, given a case c, the coverage of c are those cases in the case-base that c is able to solve correctly. The model also introduces the following property between two cases c and c': $SharedCoverage(c, c') \iff RelatedSet(c, C) \cap RelatedSet(c', C) \neq \emptyset$. Finally, these properties are used to define the following competence-based cluster criteria: a $G \subseteq C$ is a $CompetenceGroup(G) \iff \forall c, c' \in G, \exists SharedCoverage(c, c') \wedge \forall c_k \in C - G, \nexists c \in G : SharedCoverage(c_k, c)$.

In Figure2.A, we introduce a guiding example of a case-base and the basic elements of the competence model. Note that edges represent the $solves$ concept, although edges are not stored in the case-base. The example illustrates two (nested) Competence Groups obtained from CS and RS computation. Some algorithms are directly based on the Coverage-Reachability properties like COV, RFD or RC [Smyth and McKenna, 2001; Smiti and Elouedi, 2014]. Many other algorithms are inspired by the competence model. CTE [Craw *et al.*, 2007] is a redundancy reduction algorithm, as are CRR, ICF and RC. We also highlight CBE [Delany and Cunningham, 2004], a noise reduction algorithm that extends the original model, introducing the concept of liability ($LiabilitySet(c, C) = \{c' \in C \mid misclassify(c', c)\}$). All these algorithms are known as the Competence-based family. We choose RC (Algorithm 2) an *incremental* CBM process to illustrate the structure of the Competence-based family. RC algorithm uses the Relative Coverage (RC) measure $RC(c, C) = 1/|RS(c, C)|$. $OrderIncreasingRC(C)$ sorts the cases of C according to relative coverage. The complexity of such types of algorithms are, in common implementations, $O(|C|^2)$, depending on the sorting algorithm and the computation of the competence property.

[1] Many of the uncited works are referenced in [Craw *et al.*, 2007].

Algorithm 2 RC: Relative Coverage (Smyth-McKenna 2001)

Input original case-base C
Output maintained case-base C'
$L \leftarrow OrderIncreasingRC(C)$
$C' \leftarrow \emptyset$
for $c \in L$ **do**
  if c not solved in C' **then**
    $C' \leftarrow C' \cup \{c\}$ % *c competent case added: increment*
  end if
end for
return C'

3 Advances in CBM Methods

In this section, 10 methods published in the last 5 years are surveyed. We present graphical examples to illustrate and compare essential ideas. The CBM approaches are grouped as: (1) methods for improving the competence model (*volume*), (2) redefinition of the case in the case-base (*variety*), (3) the management of time in CBM (*velocity*) and (4) best solutions in complex problems (*value*). Finally, a map of key characteristics is summarised in Table 1 (criteria described in [Chebel-Morello *et al.*, 2015; Lupiani *et al.*, 2014a]).

3.1 Improving Competence

Competence-based approaches imply a high computational cost and the management of massive data (*volume*) is a challenging problem. Some authors propose variations of the model or a complete redefinition to face the problem.

Competence Measure (CM)
[Chebel-Morello *et al.*, 2015] propose a competence-based CBM algorithm in two steps. The first step (offline) calculates the coverage (Eq.(1)), the reachability (Eq.(2)), and introduces a novel competence measure (CM) for each case:

$$CM(c, C) = |CS(c, C)|/|RS(c, C)|. \quad (4)$$

Instead of using traditional RC, this new measure tries to maximize the coverage and minimize the reachability. Fig.2.B shows how $CM(c)$ is calculated from CS(c)={x,y} and RS(c)={z} continuing the example of Fig.2.A.

After *CS*, *RS* and *CM* are calculated, the algorithm categorizes each case according to such values in *auxiliary*, *support*, *spanning* and *pivotal* case labels. For instance, a case is *pivotal* when *CS=RS=CM=1*. Finally a deleting process is performed, trying to keep mainly *pivotal* cases in the case-base. The second step (online) is designed when the CBR system is running and a new case is considered to be stored. Given a new candidate, this auto-increment algorithm computes the reachability of its problem and solution and analyses the average coverage of the case-base to decide whether the new case is stored or not.

Partitioning Approaches
The CBM delete policy aims to find the worst cases to remove, chosen from the whole case-base. Unlike the general approach, partition focuses on deconstructing the case-base into subsets, treating each as an independent case-base. In [Smiti and Elouedi, 2014; 2017], the authors propose a catalogue of CBM partition algorithms sharing a basic structure: (1) a clustering algorithm is run, obtaining case-base subsets; (2) the topology of each cluster is analysed, labelling its cases; and (3) a deleting criterion is used according to the labels.

WCOID-GM algorithm is a good example of the combination of well-known Machine Learning techniques [Smiti and Elouedi, 2014]. This algorithm essentially learns the weights of the case attributes using a sample correlation technique. After that, a DBSCAN-based clustering method is used to obtain case-base subsets and univariate outlier detection uses Inter-Quartile Range methods. Finally, the central and outlier cases of each cluster are maintained and the rest are deleted.

SCBM algorithm [Smiti and Elouedi, 2017] is an evolution of WCOID-GM, including a competence model. In SCBM, a fuzzy-based DBSCAN method is used to cluster the case-base. The competence of each cluster is analysed attending to the three different types of cases: noisy cases $NC = \{c \in Cluster_k : |Cov(c)| = 0\}$, similar cases $SC = \{c \in Cluster_k : |Cov(c)| = N\}$ (N constant in $Cluster_k$) and isolated cases $IC = \{c \in Cluster_k : |Cov(c)| = 1\}$. Following the guiding example, Figure2.C shows the partition for $Cluster_i$ and its IC, SC and NC. After the type of cases are identified, SCBM removes all cases in NC and all in SC except one, while keeping all cases from IC.

Closure-Competence Model
According to [Lu *et al.*, 2014], the *CompetenceGroup* method for evaluating competence clusters (see §2.3) is inadequate and deficient. In short, the authors criticise these methods because each group may be composed of some disjoint partitions and the complete splitting is not guaranteed.

An extended competence model is proposed in [Lu *et al.*, 2014] to solve this issue, introducing 2 new concepts. First, for a given $G \subseteq C$, the Competence Closure property restricts the *CompetenceGroup* as follows:

$$CompetenceClosure(G) \iff \forall c, c' \in G, \quad (5)$$

$$\exists\, SharedCoveragePath(c, c') \wedge \quad (6)$$

$$\forall c_k \in C - G,\ \nexists c \in G : SharedCoverage(c_k, c) \quad (7)$$

We denote by $SharedCoveragePath(c, c')$ a set of $SharedCoverage(c_i, c_j)$ (see §2.3) connecting c and c'.

Features	CM	WCOID-DG	SCBM	*Pref*-CBM	FP-CA	FFD	NEFCS/SSR	T-CBM	CBNI	MOE-CBM
Approach	CaseEdit	Partition	Partition	Ftr.Edit	CaseEdit	Ftr.Edit	CaseEdit	CaseEdit	CaseEdit	CaseEdit
Direction	Dec.	Dec.	Dec.	Inc.	Inc.	Dec.	Inc.	Inc/Dec	Inc.	Inc.
Sensitivity	?	No	No	?	Yes	No	No	Yes/No	Yes	Yes
Retained	?	Cluster+Out.	Border	Cluster	?	Random	Border	Both	N/A	N/A
Determin.	Yes	No	No	Yes	Yes	No	Yes	Both	No	No

Table 1: Comparative summary of recent CBM algorithms. *Approach*: Case/Feature Editing, Partition; *Direction* of CBM process: Incremental, Decremental; *Sensitivity* to case order; Cases *Retained*: Cluster,Border,Outlier,Random; and *Deterministic.*

Second, the Related Closure extends the *RelatedSet* (Eq.(3)), where $RelatedClosure(c) = \{RelatedSet(c_i) : c \in RelatedSet(c_i)\}$.

Based on these new definitions, two competence measures are proposed: (1) a density measure to weight the Related Sets of a given Related Closure; and (2) a competence-based empirical weight to provide a reference to the degree of case distribution in a competence area (cluster).

Fig.2.D depicts an example of cases fulfilling the Competence Closure property and $RelatedClosure(c) = \{b, d, e\}$. Unlike the Competence Groups (Fig.2.A), the Competence Closure property defines disjoint sets.

3.2 Re-Structuring Competence Case-Bases

Most CBM algorithms assume a uniform case structure (e.g. vector of problem and solution) and focus on reducing the case-base size by case deletion. However, due to the need to integrate different data sources (variety), the following methods redefine this structure.

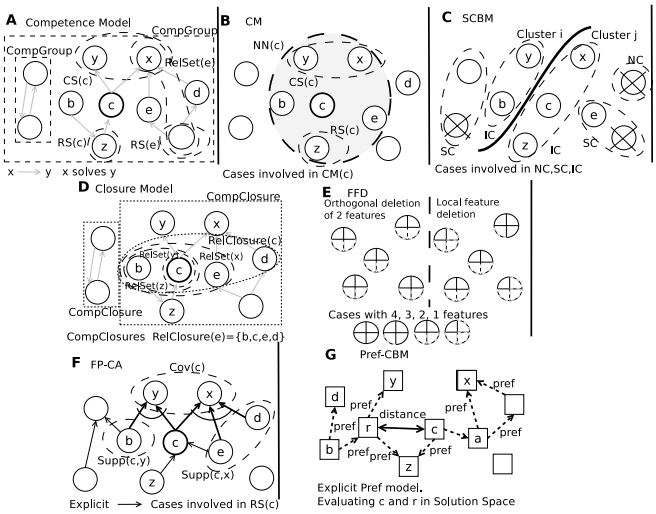

Figure 2: Comparative examples of competence CBM methods.

Flexible Feature Deletion (FFD)

In some domains, such as multimedia or social network databases, cases are large and can be represented by different levels of detail. A change in the CBM perspective is suggested in [Leake and Schack, 2015], considering maintenance at the single case level by removing part of a case; i.e. the deletion of case features causes less competence loss than removing the whole case.

The flexible feature deletion (*FFD*) approach can be used for both parts of the case (problem and solution). The core of feature deletion maintenance is, therefore, linked to case indexing in the CBR community and attribute selection in the Machine Learning field. On the one hand, case-indexing techniques traditionally study how problem attributes are displayed to increase retrieval accuracy. Unlike case indexing, feature deletion in maintenance aims to compress the case base. On the other hand, in Machine Learning attribute selection means an orthogonal deletion from the database, which is dimension reduction. *FFD* is a wider problem, including the removal of attributes of specific cases. Leake and Schack classify CBM's main strategies as case removal (traditional CBM), orthogonal feature deletion and local-feature removal.

A variety of *FFD* implementations are also presented in [Leake and Schack, 2015]. In particular, the authors experiment with removing the most common vs. most rare attribute for all cases, as well as algorithms which remove attributes randomly, for the entire case-base or some cases. Fig.2.E shows how *FFD* deletes features of 4-attribute cases (graphically as $\oplus$, one sector per feature) in the problem space. In the orthogonal deletion example the same two features are deleted in all cases while in the local feature deletion different features are removed. We believe this approach is suitable for high dimensional datasets when dimensionality reduction techniques (e.g. SVM, PCA, etc.) are not recommended.

Compositional Adaptation (FP-CA)

One key step in CBR, neglected in some CBM approaches, is the adaptation step. Essentially adaptation means that the solutions retrieved from a case-base have to be executed to be a valid solution of the new query case. [Mathew and Chakraborti, 2017] propose Compositional Adaptation (CA), a model in CBM to represent the dependency between cases to be part of a solution. In practice, this means representing the case-base using an AND-graph, where nodes are cases and edges indicate the solving capacities of single or combined cases.

They propose the *FP-CA* algorithm, a refinement of RC (Algorithm 2), changing the *RC* competence measure to the Retention Score (RSc). This new measure depends on two sets: (1) $Cov(c)$ the set of all cases in C that can be solved using c (stand-alone or in a composed solution); and (2) the $Supp(c, c_j)$, the set of cases needed to solve c_j when c is used as part of the solution. A high RSc of c means c solves many cases with a high RSc that, at the same time, they are supported by other cases with low Retention Score. From the computational point of view, the cost of the Retention Score is arguable. The case-base is a directed graph that can contain loops and therefore the Retention Score is computed for all case in an undetermined number of iterations. Formally, for iteration $k + 1$ the Retention Score (*RSc*) can be defined as:

$$RSc_{k+1}(c) = \sum_{c_i \in Cov(c)} \frac{RSc_k(c_i)}{1 + \sum_{c_j \in Supp(c,c_i)} RSc_k(c_j)).} \quad (8)$$

Following the guiding example, Fig.2.F shows how $RSc(c)$ is calculated using $Cov(c) = \{x, y\}$, $Supp(c, y) = \{b\}$ and $Supp(c, x) = \{d, e\}$. Unlike previous models (Fig.2.A-E), the case-base is an AND-graph; i.e. edges are explicitly represented in the case-base.

An empirical evaluation is carried out in [Mathew and Chakraborti, 2017], using a synthetic graph generator of case-bases, where each node c is a case and AND-edges are the $NN(c, k)$ (*k=1...4*) to compare RC and FP-CA. The authors illustrate the utility of the evaluation with a practical application for ordering the complexity of Wikipedia articles based on their semantics. In this example, each wiki page is a case $c = (\pi, \omega)$, where π is its title and ω is the meaning using the

terms of the first sentence in the article. Therefore, cases with high Retention Score are considered basic articles, while low Retention Score articles present complex concepts.

In our opinion, this model is particularly interesting when a-priori knowledge about solving capacities can be gathered or the topology of the case-base and similarities can be represented in a graph. Some examples of their applications are semantic models and social network databases.

Preference Model (Pref-CBM)

A CBM method for a novel and sophisticated version of CBR, called Pref-CBR, is presented in [Abdel-Aziz and Hüllermeier, 2015]. This model redefines the relation between a problem and its solution in a case. Instead of just having the pair (π, ω) relating a solution ω with problem π, Pref-CBR also introduces the concept of *preference*, decomposing of the case into smaller chunks of knowledge. A preference $\omega_i \succ_\pi \omega_j$ means that ω_i solution is preferred to ω_j for solving π and P_π is the set of all preferences for π. In practice, this model is implemented from a statistical perspective, considering π a random variable from a probabilistic distribution, where the parameters are estimated using Maximum Likelihood.

In [Abdel-Aziz and Hüllermeier, 2015], the CBM algorithm is a method to check whether or not a query case $c = (\pi, \omega, P_\pi)$ should be stored in the case-base. The algorithm: (1) searches in the problem space for similar cases ($KNN(c, C)$) obtaining retrieved cases ($c_r = (\pi_r, \omega_r, P_r)$); (2) for each retrieved case its solution is analysed in two steps (a) calculating the similarity between solutions using a distance function ($\delta_\omega : \Omega \times \Omega \rightarrow \mathbb{R}^+$) and (b) measuring how redundant are P_π and P_r for solving π. Finally, the query case is stored if the solution is close and the preferences are not redundant.

Fig.2.G shows an example of checking the storage of c in the case base when $KNN(c, C) = \{r\}$. Note that, unlike the rest of the examples (Fig.2.A-F), the analysis is done in the solution space ($\boxdot$ represents a solution). The preference relations ($\dashrightarrow$) between solutions are, like in FP-CA (Fig.2.F), explicilty modeled in the case-base.

The effectiveness of PrefCBM is illustrated in an experiment for solving the NP-hard Traveling Salesman Problem, evaluating the case-base size and the improvement on retrieval. This experiment considers random deleting and different configurations of PrefCBM.

In our opinion, one notable contribution is the double use of the extended solution space Ω^2. First, a distance function in the solution space is used to estimate the solution quality and, second, preferences are used as a type of heuristic.

3.3 Time in CBM

One major challenge in CBM is to provide a fast response to massive data flows (*velocity* of data processing) gathered from monitoring systems (e.g. monitoring systems). We review different perspectives about how the evolution of CBR systems over time affects CBM.

Temporal Maintenance (T-CBM)

In [Lupiani *et al.*, 2014b] we propose an extension tothe classic case structure, where a problem is a sequence of heterogeneous events, that is, $c = (< e_1, e_2, \ldots, e_n >, \omega)$ where e_i is an event occurring at time i. For such temporal cases, new distance measures are needed to manage time. In particular, we propose the use of time-point algebra and temporal editing distance, based on dynamic programming.

The maintenance of temporal cases requires the review of CBM algorithms. We propose to extend historic CBM algorithms (see §2) proposing T-CNN, T-RENN, T-DROP1-3, T-ICF and competence-based methods (T-COV, T-RC). For example, T-RENN is an extension of RENN (see Algorithm 1) where the distance function δ (traditionally Euclidean distance) is the temporal version of the edit distance. Similarly, the RC algorithm is extended, changing the sorting function *OrderIncreasingRC* using the temporal distance.

Other proposals in the CBR literature consider time series and sequences as part of the case. However, as far as we know, this is the first proposal for CBM of temporal cases.

The implemented version of T-CBM is successfully tested to maintain a temporal case-base of a CBR module to detect risk scenarios in a commercial home-monitoring system for elderly people living alone [Lupiani *et al.*, 2017]. Each problem case is the sequence of movements of a person at home during 8, 16, or 24 hours of monitoring.

Note that, in essence, the temporal extension of CBM methods keeps the structure of the original algorithm intact. We consider this fact is an advantage, since the characteristics and behaviour of the new extensions are equivalent to the original algorithms (e.g. CNN, DROP, RC) and, therefore, the new extensions can be considered mature and tested due to the wealth of studies available in the literature [Craw *et al.*, 2007]. For example, Fig.2.A can illustrate both RC and its temporal extension, keeping the RS and CS sets.

Concept-Drift-Tolerant Maintenance (Drift-CBM)

Real-world data and the goals of dynamic intelligent systems can change over time in unforeseen ways, creating the so-called concept-drift problem. Therefore, it is necessary for such systems to avoid the loss of accuracy as time passes. The Machine Learning area is very active in facing this problem, but few efforts are available in the CBM literature.

The CBM method presented in [Lu *et al.*, 2016] helps CBR systems in changing environments. Drift-CBM consists of two steps: (1) Enhancement: checks whether a new incoming case should be considered a noisy case or not if there is concept-drift (proposing NEFCS algorithm); (2) Preservation: if a storage limit exists, redundant cases are removed (proposing SRR algorithm).

Firstly, the Enhancement step focuses on a competence-based drift detection according to the *closure competence model* already surveyed (see §3.1). The approach uses the density of related sets and the competence-based empirical weight (see Eq.(1)-(3)). Fig.2.D shows an example of calculating the closure components (i.e. related sets, related closure and competence closure).

Lu et al. present NEFCS, an algorithm to prevent a novel case from being removed as noise once a competence area of the drift-concept is detected. This algorithm removes noisy cases considering the competence definition of *LiabilitySet* [Delany and Cunningham, 2004] (described in §2).

Secondly, the Preservation step removes redundant cases, proposing SRR algorithm. This algorithm combines different characteristics of historic CBM. In particular, SSR has a similar schema as CNN and follows a similar approach to IBL-DS but keeping the case-base competence (see Fig.1).

Drift-CBM and T-CBM deal with different aspects of time. While T-CBM represents the event sequences within the case, Drift-CBM supervise if the system changes over time. Drift-CBM and T-CBM algorithms are also different. Drift-CBM firstly removes noisy cases (NEFCS) and then redundant cases are deleted (SRR). In T-CBM, depending on the selected algorithm, both steps can be done at once.

3.4 Facing Computational Complexity

The complexity of CBM depends on a number of factors. According to [Smyth and McKenna, 2001], CBR systems typically operate in poorly understood, weak-theory domains. Therefore, the quality of CBM algorithms is subject to complex heuristics and, if the optimal solution exists, the search of it implies a high computational cost (*value* solution).

CBM algorithms cannot guarantee the global optimum but, in general, they effectively compute a reasonable result. Unlike CBR, which is based on lazy/online learning, Genetic Algorithms (GAs) focus on an eager data-driven strategy. GAs are good strategies to approach solutions to hard problems when there is not an analytical solution and the knowledge available is weak. GAs do not guarantee a global optimum but often converge to fitness values, considered a better approach than local optima. In recent literature we find some CBM proposals using this approach.

Case-Base Near Insertion (CBNI)

The GA-based CBR method presented in [Yamamoto *et al.*, 2015] (CBNI) is a good example of an efficient approach to solve a specific problem, in this case the Traveling Salesman Problem (TSP). The case problem is a weighted graph (representing a TSP) and the solution is a complete tour. Given a query case, former cases with similar TSPs are retrieved and a GA is run to adapt the solutions retrieved to solve the query case. The fitness function to minimise is a complete tour length of the solution.

Maintenance is therefore an essential aspect in [Yamamoto *et al.*, 2015]. In particular, three principles drive the CBM of CBNI: (1) a priori number of cases is fixed to guarantee the time of response; (2) the case-base must contain the best (shortest) solutions; and (3) the diversity of the case-base is necessary to avoid local optima.

A new solution must be added considering its fitness function but also comparing close solutions in a similar way as similarity functions in the solution space suggested by [Abdel-Aziz and Hüllermeier, 2015]. In fact, both [Abdel-Aziz and Hüllermeier, 2015] and [Yamamoto *et al.*, 2015] benchmark their CBM approaches with TSP ($O(n!)$).

Multi-Objective Evolutionary CBM (MOE-CBM)

The effectiveness of CBM algorithms depends on the proportions of noisy and redundant cases within the case-base. In [Lupiani *et al.*, 2016], we consider general CBM as a multi-objective optimization problem establishing three simultaneous objectives: (1) minimise the number of redundant cases; (2) minimise the distance to non-redundant cases; and (3) maximise the accuracy of the CBR system.

MOE-CBM is a multi-objective evolutionary CBM algorithm addressing these objectives.This algorithm is indeed an adaptation of a well-known multi-objective GA (NSGA-II). MOE-CBM explores the problem space where each problem is a potential case-base. This search is supported by the definition of noise and redundancy indicators to drive the fitness function to optimise the goals. Note that case-base size and accuracy are conflicting goals with noise and redundancy removal [Craw *et al.*, 2007].

Formally speaking, no optimal solution is guaranteed within a finite time but, in practice, the algorithm approaches acceptable solutions reducing the case-base size while maintaining the accuracy. Experiments shows MOE-CBM is the most consistent algorithm for varying conditions (noise/redundancy) of the case-base. These good results come at a cost to runtime, limiting its use to offline processes.

There are essential differences between MOE-CBM and CBNI. Firstly, from a methodological point of view, MOE-CBM follows a GA-based CBM approach while CBNI is a tailored CBM in a GA-based CBR. Secondly, unlike CBNI, MOE-CBM is a general purpose CBM. Finally, CBNI follows a redundant maintenance strategy while MOE-CBM searches for a balanced solution between redundancy and noise.

In our opinion, the key contributions of such approaches are the efforts to define what the best solution means and to find it with a reasonable computational cost.

4 Conclusion

In this work we have presented a map of CBM methods on the 50th anniversary of the first published algorithm (CNN [Hart, 1968]) and we have analysed the most recent 10 CBM algorithms. These algorithms are surveyed considering the new requirements of the massive data scenario.

We can observe that the competence-model of [Smyth and McKenna, 2001] still has a deep impact on current research. New CBM methods deal with the computation of the competence of the case-base when huge amount of data are available. We also analyse some promising models that redefine the granularity of the case-base, exploring the maintenance from different perspectives (preferences, parts of a case). We believe such methods will be suitable for high dimensional dataset from social media scenarios.

More theoretical approaches focus on searching for the optimal solution to provide the most valuable answer following a evolutionary approach. The dimension of time is also playing a key role. Some new CBM methods focus on temporal representation and dynamic systems to manage changes over time. In our opinion, these methods are helpful in modern monitoring systems.

Current CMB methods are correctly evaluated, but limited to comparisons with classic CBM algorithms. Future work should converge to a uniform evaluation methodology and this consensus might be useful in the community to fairly assess most recent results.

Acknowledgments

This work was partially funded by the MINECO Ministry under the WASPSS project (Ref: TIN2013-45491-R) and by the EFRD.

References

[Abdel-Aziz and Hüllermeier, 2015] Amira Abdel-Aziz and Eyke Hüllermeier. Case base maintenance in preference-based CBR. In Eyke Hüllermeier and Mirjam Minor, editors, *Case-Based Reasoning Research and Development*, pages 1–14. Springer, 2015.

[Chebel-Morello *et al.*, 2015] Brigitte Chebel-Morello, Mohamed Karim Haouchine, and Noureddine Zerhouni. Case-based maintenance: Structuring and incrementing the case base. *Knowledge-Based Systems*, 88:165 – 183, 2015.

[Craw *et al.*, 2007] Susan Craw, Stewart Massie, and Nirmalie Wiratunga. Informed case base maintenance: A complexity profiling approach. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*, pages 1618–1621, Vancouver, BC, 2007. AAAI Press.

[Craw, 2017] Susan Craw. Case-based reasoning. In Claude Sammut and Geoffrey I. Webb, editors, *Encyclopedia of Machine Learning and Data Mining, Edition 2*, pages 180–188. Springer US, 2017.

[Cummins and Bridge, 2011] Lisa Cummins and Derek Bridge. Choosing a case base maintenance algorithm using a meta-case base. In *Proceedings of the 28th SGAI Intl. Conference on Innovative Techniques and Applications of Artificial Intelligence*, pages 167–180, Cambridge, UK, 2011. Springer.

[Delany and Cunningham, 2004] Sarah Jane Delany and Pádraig Cunningham. An analysis of case-base editing in a spam filtering system. In Peter Funk and Pedro A. González Calero, editors, *Advances in Case-Based Reasoning*, pages 128–141, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[Goel and Diaz-Agudo, 2017] Ashok K. Goel and Belen Diaz-Agudo. What's hot in case-based reasoning. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 5067–5069, 2017.

[Hart, 1968] P. Hart. The condensed nearest neighbor rule. *IEEE Transactions of Information Theory*, 14(3):515–516, May 1968.

[Leake and Schack, 2015] David Leake and Brian Schack. Flexible feature deletion: Compacting case bases by selectively compressing case contents. In Eyke Hüllermeier and Mirjam Minor, editors, *Case-Based Reasoning Research and Development*, pages 212–227. Springer, 2015.

[Lu *et al.*, 2014] Ning Lu, Guangquan Zhang, and Jie Lu. Concept drift detection via competence models. *Artificial Intelligence*, 209:11–28, April 2014.

[Lu *et al.*, 2016] Ning Lu, Jie Lu, Guangquan Zhang, and Ramon Lopez de Mantaras. A concept drift-tolerant case-base editing technique. *Artificial Intelligence*, 230:108 – 133, 2016.

[Lupiani *et al.*, 2014a] Eduardo Lupiani, Jose M. Juarez, and Jose Palma. Evaluating case-base maintenance algorithms. *Know.-Based Syst.*, 67:180–194, September 2014.

[Lupiani *et al.*, 2014b] Eduardo Lupiani, Jose M. Juarez, and Jose Palma. A proposal of temporal case-base maintenance algorithms. In Luc Lamontagne and Enric Plaza, editors, *Case-Based Reasoning Research and Development*, pages 260–273. Springer, 2014.

[Lupiani *et al.*, 2016] Eduardo Lupiani, Stewart Massie, Susan Craw, Jose M. Juarez, and Jose Palma. Case-base maintenance with multi-objective evolutionary algorithms. *Journal of Intelligent Information Systems*, 46(2):259–284, Apr 2016.

[Lupiani *et al.*, 2017] Eduardo Lupiani, Jose M. Juarez, Jose Palma, and Roque Marin. Monitoring elderly people at home with temporal case-based reasoning. *Knowledge-Based Systems*, 134:116 – 134, 2017.

[Mathew and Chakraborti, 2017] Ditty Mathew and Sutanu Chakraborti. Competence guided model for casebase maintenance. In *Proceedings of the 26th Intl. Joint Conference on Artificial Intelligence*, pages 4904–4908, 2017.

[Salamó and Golobardes, 2001] Maria Salamó and Elisabet Golobardes. Rough sets reduction techniques for case-based reasoning. In David W. Aha and Ian Watson, editors, *Case-Based Reasoning Research and Development*, pages 467–482. Springer, 2001.

[Smiti and Elouedi, 2014] Abir Smiti and Zied Elouedi. WCOID-DG: An approach for case base maintenance based on weighting, clustering, outliers, internal detection and Dbsan-Gmeans. *Journal of Computer and System Sciences*, 80(1):27 – 38, 2014.

[Smiti and Elouedi, 2017] Abir Smiti and Zied Elouedi. SCBM: Soft case base maintenance method based on competence model. *Journal of Computational Science*, 2017.

[Smyth and Keane, 1995] Barry Smyth and Mark T. Keane. Remembering to forget: A competence-preserving case deletion policy for case-based reasoning systems. In *Proceedings of the 14th Intl. Joint Conference on Artificial Intelligence - Volume 1*, pages 377–382, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[Smyth and McKenna, 2001] Barry Smyth and Elizabeth McKenna. Competence models and the maintenance problem. *Computational Intelligence*, 17(2):235–249, 2001.

[Tomek, 1976] Ivan Tomek. An experiment with the edited nearest-neighbor rule. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-6(6):448–452, June 1976.

[Yamamoto *et al.*, 2015] Y. Yamamoto, T. Kawabe, Y. Kobayashi, S. Tsuruta, Y. Sakurai, and R. Knauf. A refined case based genetic algorithm for intelligent route optimization. In *11th Intl. Conference on Signal-Image Technology Internet-Based Systems (SITIS)*, pages 698–704, Nov 2015.