An evidential integrated method for maintaining case base and vocabulary containers within CBR systems

Safa Ben Ayed, Zied Elouedi, Eric Lefevre

An Evidential Integrated Method for Maintaining Case Base and Vocabulary Containers within CBR Systems

Safa Ben Ayed[a,b,*], Zied Elouedi[a,*], Eric Lefevre[b,*],

[a]*LARODEC, Institut Supérieur de Gestion de Tunis, Université de Tunis, Tunis, Tunisia*
[b]*LGI2A, Univ. Artois, EA 3926, 62400 Béthune, France*

Abstract

Cases and vocabulary maintenance presents a crucial task to preserve high competent Case-Based Reasoning (CBR) systems, since the accuracy of their offered solutions are strongly dependent on stored cases and their describing attributes quality. The maintenance aims generally at eliminating two types of undesirable knowledge which are noisy and redundant data. However, inexpedient Case Base Maintenance (CBM) or vocabulary maintenance may not only greatly decrease CBR competence in solving new problems, but also reduce its performance in term of retrieval time. Besides, to provide a high maintenance quality, it is necessary to manage uncertainty within knowledge since "real-world data are never perfect" and stored cases within a CBR system's Case Base (CB) describe real-world experiences. Hence, we propose, in this paper, a new integrated method that maintains both of the CB and the vocabulary knowledge containers of CBR systems by offering a new alternating technique to properly detect noisiness and redundancy whether in cases or features. During the learning steps of our new integrated maintenance policy, which drives the decision making about cases and attributes selection, we manage uncertainty using one among the most powerful tools called the Belief Function Theory.

Keywords: Case-Based Reasoning, Case Base Maintenance, Vocabulary Maintenance, Machine Learning, Uncertainty, Belief Function Theory.

1. Introduction

In CBR systems, reasoning is quite special comparing to reasoning in databases and logic [1]. It can be described as a mimicking of the human reasoning way to solve new problems. The basic operating assumption in CBR is that "Similar problems have similar solutions", which makes CBR a kind of approximate reasoning. Basically, CBR systems call past recorded experiences to solve new problems and make decisions. The set of these experiences is stored in form of problem-solution couples in a memory structure called a Case Base (CB). To solve a new problem, the well-known *4R* cycle of traditional CBR [5] can be applied. First, the CBR system *R*etrieved from the CB the closest case(s) to the current problem using a similarity measure and the vocabulary describing cases. Second, the *R*euse phase consists at applying some adaptations to the solution of the retrieved case to fit the current problem. In case of rejection to the proposed solution (e.g., by a domain expert), it should be *R*evised. Finally, the new problem with its solution are *R*etained in the CB as a new case, and the system is supposed to boost its competence in problem resolution. This ability to learn incrementally makes that kind of reasoning widely applied in several contexts such as medicine and health area [2], finance [3], design and manufacturing [4], etc. However, after a given period of time, we can note some decreasing of the competence and/or the performance of the system, which is originated mainly of two causes. First, the emergence of noisy cases inside the CB which can introduce negative impact. Second, the emergence of a large number of redundant cases that cover the same problem space and degrades, therefore, the

*Corresponding author
Email addresses: benayedsafa@gmail.com (Safa Ben Ayed), zied.elouedi@gmx.fr (Zied Elouedi), eric.lefevre@univ-artois.fr (Eric Lefevre)

system's performance in term of retrieval time. The two latter discussed problems are massively tackled under the Case Base Maintenance (CBM) [6] field, where a large number of policies aim to clean up CBs from irrelevant cases.

Obviously, the CB is the basic and essential component of any CBR system. However, if we look to CBR from the knowledge viewpoint, we distinguish different repositories containing distinct varieties of knowledge [1] that should be given some attention. In CBR, these knowledge are stored in four knowledge containers [9]: The Case Base, the vocabulary, the similarity measures, and the adaptation knowledge (Figure 1).

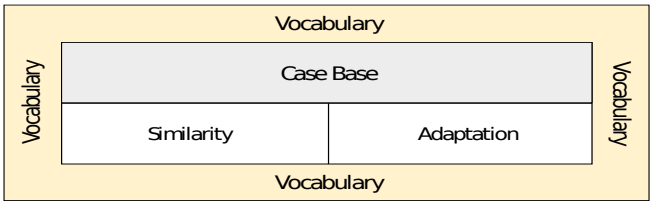

Figure 1: The four knowledge containers of CBR

In addition to CBs, the vocabulary knowledge plays an important role in offering relevant solutions. Actually, vocabulary container holds knowledge that describe explicitly used elements [1]. For instance, in the context of Structural CBR systems[1] where cases are described in form of attribute-value pairs, we can suppose that the vocabulary is restricted to the set of attributes describing cases. Actually, in every real-world experience, there is a really huge number of attributes that can describe it. However, only some of them are relevant for the decision making about a particular task. For that reason, the vocabulary container presents also the subject of a considerable maintenance target [10] to provide a competent and up-to-date CBR system.

In the context of these requirements, we propose in this paper an integrated method that maintains simultaneously both of the CB and the vocabulary knowledge containers by eliminating noisiness and redundancy in both cases and attributes. Our objective behind this work is to maintain efficiency and competence of CBR systems by keeping only accurate and relevant attributes and cases. However, it is obviously easier said than done, especially that cases descriptions are not quite precise and exact. This imperfection in data makes birth of the necessity in managing, measuring and minimizing uncertainty. Since this uncertainty can appear in different aspects and with graduated levels, we make use, during the learning steps of our new maintaining method, some powerful managing uncertainty tools offered in the frame of the belief function theory [11, 12].

The remind of this paper is organized as follows. The next Section recalls the fundamental concepts of the belief function theory, as well as the two used evidential machine learning techniques. Section 3 focuses on presenting the field of Case-Based Reasoning Maintenance along with the related work. The details about our new integrated method named IMMEP *(Integrated Maintaining Method based on Evidential Policies for CBR maintenance)* as well as its algorithm are presented throughout Section 4. Thereafter, we establish, during Section 5, an experimental study followed by results exposition and discussion. Conclusions and outlook are finally stated in Section 6.

2. Belief Function Theory

As our contribution, for this paper, aims to manage the uncertainty involved in past experiences while performing a two-dimensional maintenance task (CB and vocabulary), we provide, in Section 2.1, the fondamental concepts of the belief function theory, which is named also Demspter-Shafer or Evidence theory [11, 12]. During Section 2.2, the concept of credal partition is defined to model the doubt about cases assignment to clusters within this theory. For the learning steps, the two used evidential clustering algorithms are presented respectively in Sections 2.3 and 2.4.

2.1. Fundamental Concepts

To model and quantify the evidence under the framework of the belief function theory, we consider ω as a variable referring to K elementary events to a given problem defined by $\Omega = \{\omega_1, \omega_2, ..., \omega_K\}$, called the *frame of discernment*.

[1] There is also Conversational CBR and Textual CBR which differ by their way in representing cases [1].

The power set of that problem is defined by the set of all the 2^K possible subsets of the predefined events taking values in Ω. The key point of the belief function theory is the basic belief assignment (bba) which represents the item of evidence (partial knowledge) regarding the actual value of ω. It is defined as a function m from 2^Ω to $[0, 1]$ verifying $\sum_{A\subseteq\Omega} m(A) = 1$. If $m(A) > 0$, then A is called *focal element*. The basic belief mass $m(A)$, which denotes the belief degree assigned to the hypothesis "$\omega \in A$", can be attached to a subset of variables regardless any additive assumption. A bba corresponds to the open world assumption [13], if we allow evidence assignment to the empty set. This means that Ω is incomplete and ω can be taken outside. This interpretation is meaningful especially in clustering when we aim to detect noises [14]. Contrariwise, the bba function corresponds to the closed-world assumption [12] if it is normalized ($m(\emptyset) = 0$).

Among the functions that can be computed through the bba, we cite the plausibility which presents the maximum amount of belief supporting a subset A, and it is defined as follows:

$$pl(A) = \sum_{B\cap A\neq\emptyset} m(B) \qquad \forall A \subseteq \Omega \tag{1}$$

If we consider two mass functions m_1 and m_2 defined within the same frame of discernment, then we can compute the degree of conflict with several manners. One among the most known methods is defined, in [12], such that: $\kappa = \sum_{A\cap B=\emptyset} m_1(A)\, m_2(B)$. Last but not least, to make decision about the best hypothesis regarding a normalized bba m, we can use the pignistic probability transformation defined within the TBM framework [13] as follows:

$$BetP(\omega) = \sum_{\omega\in A} \frac{m(A)}{|A|} \qquad \forall \omega \in \Omega \tag{2}$$

2.2. Credal Partition

Managing uncertainty using the belief function theory within clustering problem has been largely applied. The clustering is a machine learning technique aiming at revealing some structures on data by organizing objects according to their similarity. The more the objects are somehow similar, the more probable to belong to the same group.

Under the evidential clustering problem, the concept of *Credal Partition* is built through assigning a belief degree of membership not only to singletons of the frame of discernment, but also to all possible subsets of clusters. Actually, the frame of discernment, within the evidential clustering frame, refers to the set of K possible clusters. The partial knowledge regarding the membership of an object o_i to a partition of clusters A is defined by a mass function denoted $m_i(A)$. Its value supports the hypothesis "The real cluster of object o_i belongs to the partition A". That bba quantifies the uncertainty regarding the membership of only one object. If we have n objects, then the credal partition presents the set of $n - tuple$ bbas $(m_1, m_2, ..., m_n)$.

Thereafter, we present two evidential clustering techniques that are used in our contribution to quantify uncertainty through their generated credal partitions: The first is object-based (used for CB maintenance) and the second is dissimilarity-based (used for vocabulary maintenance).

2.3. Evidential Object Clustering: Evidential C-Means (ECM)

Evidential C-Means (ECM) [14] is an object clustering method that manages the uncertainty of their membership to clusters and generates the credal partition. Within ECM, each cluster ω_k is presented by its center $\boldsymbol{v_k}$ in form of data features vector. Therewith, a partition A_j of clusters is presented by a prototype $\overline{v_j}$, which is defined by the barycenter of the different prototypes of each single cluster in A_j, as follows:

$$\overline{\boldsymbol{v_j}} = \frac{1}{|A_j|} \sum_{k=1}^{K} s_{kj} \boldsymbol{v_k} \tag{3}$$

where $s_{kj} = 0$ if $\omega_k \notin A_j$ and $s_{kj} = 1$ otherwise, with K presenting the total number of clusters.

The main idea of ECM algorithm is to generate the credal partition by minimizing the following cost function:

$$J_{ECM}(M, V) = \sum_{i=1}^{n} \sum_{j/A_j\neq\emptyset, A_j\subseteq\Omega} |A_j|^\alpha m_{ij}^\beta d_{ij}^2 + \sum_{i=1}^{n} \delta^2 m_{i\emptyset}^\beta \tag{4}$$

subject to

$$\sum_{j/A_j \subseteq \Omega, A_j \neq \emptyset} m_{ij} + m_{i\emptyset} = 1 \qquad \forall i = 1...n \tag{5}$$

where n is the data size, M presents the space of the credal partition, V presents the space of prototypes, m_{ij} denotes $m_i(A_j)$, and d_{ij} is the euclidean distance between object o_i and the partition A_j. The parameter α aims at controlling the penalization degree for partitions having high cardinality, while β and δ treat noisy objects.
More details about the minimization of the cost function J_{ECM} can be found in [14].

2.4. Evidential dissimilarity-based Clustering: k-EVCLUS method

Evidential dissimilarity clustering methods aim to generate the credal partition regarding objects membership to clusters not through their attributes values, but through the relation between them in term of dissimilarity. EVCLUS [15] is one among the most known relational evidential clustering methods. In this paper, we make use of an improvement of that technique called k-EVCLUS [16][2] which makes the original algorithm more able to handle larger dissimilarity data.

Let consider $D = (d_{ij})$ a square matrix of size n containing the dissimilarities between n objects. The main idea of k-EVCLUS is to generate the credal partition M in such a way that the conflict degrees κ_{ij} between any two bbas m_i and m_j[3] fit the dissimilarities d_{ij} between objects o_i and o_j. By this way, the credal partition is generated by assigning mass functions with low (respectively high) conflict to similar (respectively dissimilar) objects. k-EVCLUS consists, therefore, at minimizing, for only a subset of pairs of objects (o_i, o_j), the sum of the square error terms $(\kappa_{i,j} - \varphi(d_{ij}))^2$, where $\varphi(d) = 1 - exp(-\gamma\, d^2)$ and γ can be fixed to $-log\alpha/d_0^2$, with a recommendation to fix α to 0.005 and d_0 to some quantile of dissimilarities d_{ij}. If the number of clusters is large, it was recommended to assign a small value to d_0. To be done, some value of $k < n$ is chosen[4] and, for each object o_i, k other objects $(j_1(i), j_2(i), ..., j_k(i))$ are randomly selected to minimize, over the object pairs $(o_i, o_{j_r(i)})$, the following stress function:

$$J_k(M) = \eta \sum_{i=1}^{n} \sum_{r=1}^{k} (\kappa_{i,j_r(i)} - \delta_{i,j_r(i)})^2 \tag{6}$$

where $\delta_{i,j_r(i)} = \varphi(d)$, with d is the dissimilarity between object o_i and object $o_{j_r(i)}$.

3. Case-Based Reasoning Maintenance (CBRM)

Actually, CBR systems are designed to work for long time frame. Hence, some maintenance operations to their knowledge containers are needed between-whiles, and this need remains as well as the system survives. We propose, therefore, modeling this idea by the loop shown in Figure 2.

In this work, we focus, on the maintenance of the CB and the vocabulary since we consider the *Null-Adaptation* [17] and apply an automated update when calculating the similarity according to the remaining features after maintenance. Hence, during the rest of this Section, we will pay attention to the CB and vocabulary maintenance.

3.1. Vocabulary Maintenance

What is Vocabulary? As shown in Figure 1, the vocabulary container is the basis of all the other three knowledge containers [18]. It responses to the question *"Which elements of the data structures are used to present fundamental notions?"* [19]. Actually, the vocabulary depends on the nature of knowledge sources which can be attribute-value data with an object-oriented organization[5] or more complex data types such as text, image, sensor data, speech, etc.

[2] https://CRAN.R-project.org/package=evclust
[3] In other term, the plausibility to belong to the same cluster.
[4] It presents the origin of the letter k within the term k-EVCLUS.
[5] The most common structure in CBR systems.

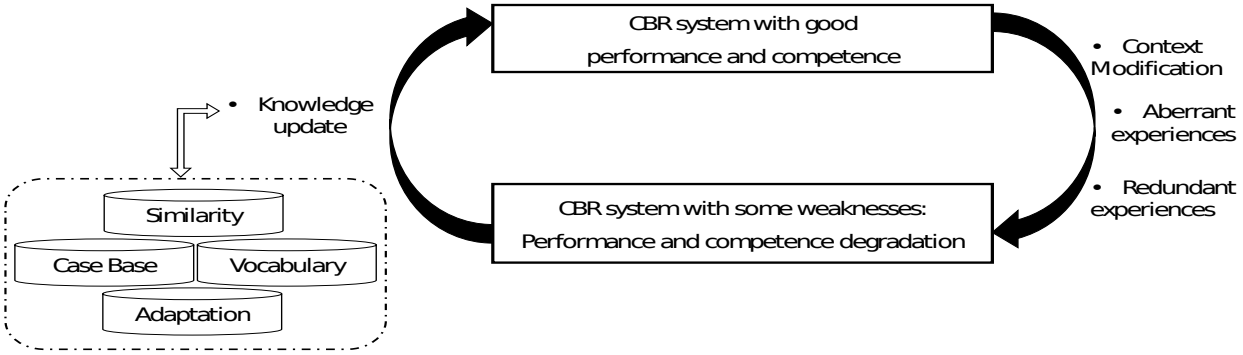

Figure 2: CBR systems maintenance

Hence, the vocabulary can be attributes, predicates, functions, set of words, and related constructs. For our purpose in the current work, it is sufficient to restrict the vocabulary on the set of attributes[6] describing cases.

Why maintenance? As mentioned in the introduction, an experience can be described by an infinite number of features with different levels of granularity. However, only few of them conduct to the accurate decision. We can defend the necessity of maintaining the vocabulary from two points of view. Firstly, the redundancy of attributes can overburden the retrieval process without any added-value in term of competence for problem solving. Secondly, noisy features prevent the overall systems to be conducted to the good solution. For the redundancy in attributes, it can be presented as a kind of high-correlated features where the deletion of some of them does not influence the whole system's capability of decision making. For noisy attributes, they present the set of features that their deletion conducts to the improvement of the system's decision making accuracy.

Why uncertainty management? According to authors in [20], the vocabulary as well as the three other knowledge containers are the origins of uncertainty in CBR systems. In fact, we cannot handle vocabulary and maintain the set of attributes without well analyze the set of values describing each one. Since these values are never exact, they are undoubtedly full of uncertainty and imprecision, which gives birth to the vital need to manage this imperfection during the learning steps at the aim of making the most accurate decision.

Related work. Actually, the maintenance of features is widely explored within the context of *Feature Selection* (FS)[7] or *Feature Reduction* (FR). Hence, we find, in the literature, several works that select, reduce or delete features describing cases to ensure accurate retrieval outcomes, such that in [10, 21, 22, 23]. One among the useful concepts to select relevant features within CBR is the *Attribute Clustering* which has been carried out, for that matter, in several works [10, 24, 25]. Similarly to object clustering, features belong to the same cluster are somehow similar. Conversely, dissimilar attributes should belong to different clusters. Basically, features similarity reflects the relation between them. It can actually be in term of correlation, dependency, etc[8].

Discussion. While maintaining vocabulary for CBR systems, it is important to take into account and preserve the relation between cases features. That's why, recommendations are offered [10, 24] to regroup features during the learning step in order to eliminate redundant and irrelevant attributes. Besides, attribute clustering provides a flexibility to CBR systems by offering the possibility to replace any feature by another one belonging to the same cluster. However, existing policies aiming to maintain CBR vocabulary are neither able to preserve the relation between attributes nor to manage the imperfection within data.

[6] During the rest of this paper, we use *attribute* and *feature* terms exchangeably.
[7] It is an NP-Hard problem.
[8] The choice of this relation depends basically on the research goal.

3.2. Case Base Maintenance (CBM)

The CB, that stores the set of past experiences to be retrieved, is the basic element of any CBR system. Since CBR systems differ from the other knowledge-based systems in their way of reasoning, and especially in their ability to learn incrementally, the maintenance of case bases becomes essential to revise the CB organization [6]. For that reason, CBM is more explored in-depth, and we find, in the literature, a considerable number of CBM policies that target CBR systems for some performance objectives. For instance, numerous CBM policies revise CB's content through the selection of only representative cases that are able to cover the remaining set of cases' problems such as the Condensed Nearest Neighbor (CNN) [7] which represents the baseline of data reduction methods. Its idea is to iteratively and randomly select cases and add them in a new CB. During each iteration, CNN tests if the new CB is able to successfully solve all problems in the original one. If not, a new iteration is carried out. Otherwise, CNN stops iterations and the new maintained CB is totally built. We cite, moreover, the Reduced Nearest Neighbor (RNN) [8] which initializes its maintained CB to the original one, then reduce it case by case while no case from the original CB is misclassified by the reduced one.

Besides, we find other CBM policies [30, 31, 32, 33] that apply maintenance operations according to some evaluation criteria such that *(1)* the performance which is quantified by the time spent to solve a target problem, and *(2)* the competence which represents the range of problems that the CB can successfully solve [26, 29].

Some CBM are based on partitioning CBs which gives the ability to treat the original CB in form of small ones. Actually, cases clustering is widely used within the CBM field due to its approved utility in detecting cases to be maintained. For instance, Clustering, Outliers and Internal cases Detection method (COID) [34] performs the density based clustering technique called DBSCAN [35] and, then, selecting only cases that their deletion affects the whole CB quality. A variant of COID method called WCOID-DG [36], for Weighting, Clustering, Outliers, Internal Detection and Dbsan-Gmeans based policy, is established by combining a weighted version of COID with the Gaussian-means clustering technique so as to well fix the number of clusters. Besides, authors in [37] propose a clustering based deletion policy that exploits the K-Means as a clustering technique.

However, all the mentioned policies are not able to manage imperfection within knowledge. Therefore, we cite others, that aim to tackle the problem of uncertainty management within CBM. SCBM method [38], for instance, is one among the soft partitioning based CBM policies that is based on the fuzzy logic [28] and uses the Soft DBSCAN-GM (SDG) [39] as a clustering technique. On the other hand, the policies Evidential Clusterering and case Types Detection for CBM (ECTD) [40], Dynamic policy CBM based on EK-NNclus algorithm and case Types Detection (DETD) [41] and Constrained Evidential Clusterering and case Types Detection for CBM (CECTD) [42] use the belief function theory tools [11, 12] (see Section 2) to manage uncertainty for cases partitioning as well for the decision about cases removal. The main maintenance principle of all these policies consists at classifying cases into a number of types according to their characteristics regarding the overall CB's competence (see Section 4.2).

We remark, after this study, numerous works interested in maintaining CBR systems, which expresses their significance in real applications. However, no work is perfect and some weaknesses are noted. Therefore, we tackle some of them by proposing, in the following Section, a new integrated maintaining approach that is characterized by *(1)* an alternation between Cases and Vocabulary learning, *(2)* an uncertainty management within both CB and vocabulary, and *(3)* a removal of redundant and noisy cases with a selection of only relevant and representative attributes.

4. Integration of CBM and Vocabulary Maintenance Policies within the Belief Function Framework

The overall outcome of CBR systems is affected by the maintenance operations. Hence, we aim to repair CBR systems weaknesses (Figure 2) by proposing an integrated maintenance policy that targets CBR systems knowledge, named *IMMEP* for *"Integrated Maintaining Method based on Evidential Policies"*. All concepts and tools offered by the belief function theory[9] (Section 2) can only indicate its richness and flexibility to be used in different fields. For that reason, we choose to use it, within our new IMMEP method, for knowledge uncertainty management, during the maintenance of CBR systems. As mentioned in the introduction, we aim, in this paper, to eliminate noisiness and redundancy in both of cases and attributes. The intuitive idea that arises is to perform successively a CBM method then a vocabulary maintaining method. However, in that case, we will use, during cases learning, the set of features

[9] A lot of other tools within this theory are not mentioned in this paper.

that probably contains some noisy attributes which distort CB learning. Similarly, the learning of features is done through using cases description (attributes values) that, also, can be characterized by some distortions. Actually, this can negatively affect the accuracy of maintenance operations whether in attributes level or cases level. Therefore, a number of alternations between two main learning and maintaining phases is followed by our new IMMEP method:

- The first phase, that is inspired from our preliminary work described in [10], concerns vocabulary maintenance.
- The second phase regarding case base maintenance uses steps of a new weighted version of our preliminary work described in [40].

In what follows, we describe, therefore, our vocabulary maintenance strategy (Section 4.1), our new weighted policy for CBM (Section 4.2), and our integrated IMMEP method for CB and vocabulary maintenance (Section 4.3).

4.1. Vocabulary maintenance with uncertainty management: Evidential Vocabulary Maintenance policy (EVM)

As already mentioned, we aim to eliminate noisy and redundant features. To do, we use a machine learning technique that deals with this problem and able to manage uncertainty. In our context, we call attributes as redundant if they are highly correlated, since they offer the same information. From the other side, noisy features distort the CBR system's outcome and their elimination improves the supplied solution's accuracy. As shown in Figure 3, the vocabulary knowledge is used in form of an ensemble of features to describe the set of cases. For instance, if the vocabulary is described in term of p attributes $\Gamma = \{A_1, A_2, ..., A_p\}$, then the description of one case x_i within the vocabulary Γ is defined by $\{a_{i1}, a_{i2}, ..., a_{ip}\}$. In the current research work, these values are considered as numeric to reach our four steps for vocabulary maintenance as detailed in Figure 3 and during the following Subsections.

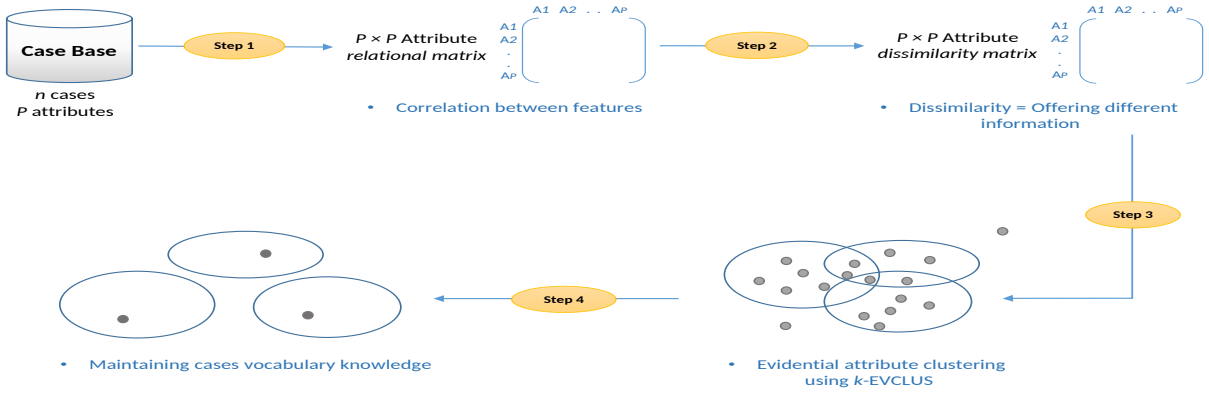

Figure 3: The mains steps of our vocabulary maintenance strategy

4.1.1. Step 1: Relational matrix generation

In our method, relation between features reflects their amount of correlation. Hence, given a CB containing n cases described by p features, we define $R = (r_{A_iA_j})$ as attributes relational matrix where $r_{A_iA_j}$ measures the linear association between the two features A_i and A_j using the *Pearson's correlation coefficient* [43], which is defined as follows:

$$r_{A_iA_j} = \frac{\sum_{l=1}^{n} (a_{il} - \overline{a_i})(a_{jl} - \overline{a_j})}{\sqrt{\sum_{l=1}^{n} (a_{il} - \overline{a_i})^2} \sqrt{\sum_{l=1}^{n} (a_{jl} - \overline{a_j})^2}} \tag{7}$$

where a_{il} (respectively a_{jl}) represents the different values of attribute A_i (respectively A_j) for case l, and $\overline{a_i}$ and $\overline{a_j}$ are their mean values.

4.1.2. Step 2: Dissimilarity matrix generation

Two features are said to be similar if they are highly correlated. According to this idea, the dissimilarity matrix $D = (d_{A_iA_j})$ is calculated in function of attributes correlations, which are bounded between -1 and 1. By this way, the following three main situations arise:

- *Situation* 1: If $r_{A_iA_j} \simeq 1 \Rightarrow$ High correlation (positive) $\Rightarrow$ Similar provided information $\Rightarrow$ High similarity.
- *Situation* 2: If $r_{A_iA_j} \simeq -1 \Rightarrow$ High correlation (negative) $\Rightarrow$ Similar provided information $\Rightarrow$ High similarity.
- *Situation* 3: If $r_{A_iA_j} \simeq 0 \Rightarrow$ No correlation $\Rightarrow$ Different provided information $\Rightarrow$ High dissimilarity.

Consequently, the similarity between two features A_i and A_j is equal to $|r_{A_iA_j}|$ and their dissimilarity is, therefore, calculated such that: $d_{A_iA_j} = 1 - |r_{A_iA_j}|$.

4.1.3. Step 3: Evidential attribute clustering

At this step, we aim to regroup features through the generated dissimilarity matrix by using an evidential dissimilarity based clustering technique. For that, we use k-EVCLUS (presented in Section 2.4) which is a recent and powerful technique for that matter. Its ability to regroup cases into clusters with managing uncertainty, as well as in detecting outliers, eases the next task regarding maintenance.

4.1.4. Step 4: Features maintenance

Ultimately, this step ensures vocabulary maintenance by eliminating usefulness and irrelevant features. Hence, it is divided into three main sub-steps summed up as follows:

- Removing noisy features, detected during the previous step, since they can seriously decrease the problem-solving competence.
- Making decision about attributes membership to clusters, from the credal partition, using the Pignistic probability transformation (Equation 2).
- Selecting only one representative feature from each cluster and eliminating all the redundant attributes.

4.2. CBM with uncertainty management: Weighted version of ECTD (WECTD)

Aiming to select relevant cases from the CB, our strategy performs a new weighted evidential clustering and detects four types of cases according their competence in covering problem-space. They are briefly defined such as:

- Noisy cases: They represent the set of cases with distortion of values that, logically, cannot belong to the CB.
- Similar cases: The set of cases that have similar descriptions presents the majority of experiences in a CB and, hence, considered as redundant.
- Isolated cases: They are somehow different comparing to the majority of cases. This set of cases is important to cover non-common problems.
- Internal cases: Each internal case represents the prototype of each group of similar cases. They aim at covering all redundant cases.

The new weighted evidential clustering consists at developing a weighted version of the Evidential C-Means clustering technique (Section 2.3). It serves mainly at managing features competence within learning by affecting a normalized weight for every attribute describing cases. This version, hence, does not believe that the set of features has the same degree of relevance. The more a weight is near to one, the more its corresponding attribute offers valuable information in solving problems. Actually, these weights regarding the importance or relevancy of features may be provided directly by domain-experts or also elicited in different ways like using feature weighting algorithms as presented in [44]. For instance, in the main contribution of this paper, weights have been elicited through the mix between initialization and penalization (Section 4.3). A general depict of this strategy with the different steps are illustrated in Figure 4, where more details about each step are given within the following Subsections.

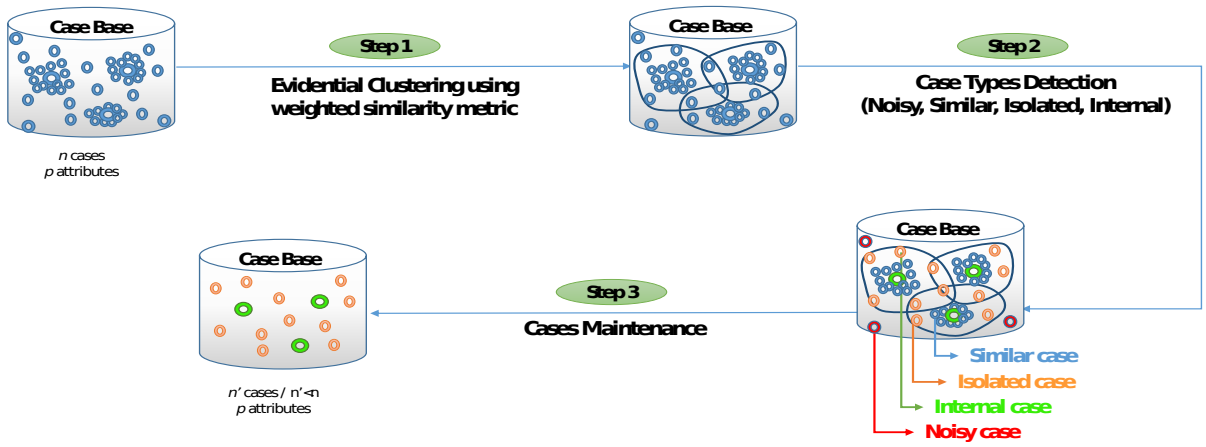

Figure 4: The mains steps of WECTD policy

4.2.1. Step 1: Evidential clustering of cases using weighted metric

First of all, our CBM maintenance policy aims to perform a learning step on the set of cases. Therefore, this step consists at regrouping cases with membership uncertainty management, since cases contain a high amount of inaccuracy. In our preliminary work [40], the default version of Evidential C-Means (ECM) [14] has been used. However, the latter evidential clustering technique uses the basic euclidean distance during the optimization of its cost function. This metric does not allow assigning more importance and significance to more relevant features in solving problems than the others. At the aim of targeting the main objective of this paper in building an integrated vocabulary and CB maintenance method, we developed a Weighted version of ECM (WECM) to be used within our new IMMEP method (Section 4.3). The difference between ECM and WECM is the way in calculating distances and the use of weights. Consequently, WECM presents the first step of our WECTD policy, which is used as the evidential clustering technique that aims to generate the credal partition regarding cases membership to partitions of clusters, as well as partitions' centers (see Section 2.3).

4.2.2. Step 2: Case types detection

The idea behind dividing a CB into some case types has been used in several works such in [27, 34, 36, 38, 40, 41]. This categorization is generally based on two fundamental concepts related to the CB's competence called Case reachability and Case coverage [29]. Hence, we detect our four types of cases based on the following definitions [40]:

Definition 4.1. A case x_i is said reachable by a CB if it contains at least one case similar or close to x_i.

Definition 4.2. Each case belonging to a set of similar cases is able to cover all the other cases set items.

Noisy cases detection. Noisy cases can be detected in the same way as in [45]. The idea consists at allocating a cluster to which noises will be assigned. Actually, ECM clustering technique, which is applied during the first step, assigns noises, that cannot be assigned to any one among clusters, to the empty set partition. By this way, we detect noisy cases as those having a membership degree to belong to the empty set partition higher than to belong to all the other partitions. It is defined as follows:

$$\boldsymbol{x_i} \in NC \;\; iff \;\; m_i(\emptyset) > \sum_{A_j \subseteq \Omega, A_j \neq \emptyset} m_i(A_j) \tag{8}$$

where $\boldsymbol{x_i}$ presents one case from the CB and NC represents the set of Noisy Cases.

Similar and Isolated cases detection. Without considering cases already detected as noises, we categorize the remaining as Similar or Isolated through measuring distances between cases and partitions centers. In fact, Similar cases are situated around partitions centers, and isolated cases are found in their borders. Hence, the distinction between them is done, foremost, after measuring cases distance to clusters, and comparing them, afterwards, to a threshold, which is defined by the mean of distances.
To manage uncertainty, also, in distance measuring, we take advantage of the generated credal partition and use the Belief Mahalanobis Distance [40] defined such that:

$$BMD(\boldsymbol{x_i}, \boldsymbol{v_k}) = \sqrt{(\boldsymbol{x_i} - \boldsymbol{v_k})^T \Sigma_k^{-1} (\boldsymbol{x_i} - \boldsymbol{v_k})} \tag{9}$$

where $\boldsymbol{v_k}$ represents the center of cluster ω_k and Σ_k is the *Belief Covariance Matrix* of the k^{th} cluster which has been defined, in [46], as follows:

$$\Sigma_k = \sum_{i=1}^{n} \sum_{A_j \ni \omega_k} m_{ij}^2 |A_j|^{\alpha-1} (\boldsymbol{x_i} - \overline{\boldsymbol{v_j}})(\boldsymbol{x_i} - \overline{\boldsymbol{v_j}})^T \qquad \forall k = 1, .., K \text{ and } A_j \subseteq \Omega \tag{10}$$

where K represents the total number of clusters, m_{ij} and $\overline{\boldsymbol{v_j}}$ present respectively the credal partition and their prototypes as defined during the evidential clustering step. The parameter α serves to penalize the belief's allocation to partitions with high cardinality.

The distinction between similar and redundant cases is, hence, achieved as follows:

$$\boldsymbol{x_i} \in \begin{cases} SC_k & if\ \exists k / BMD(\boldsymbol{x_i}, \boldsymbol{v_k}) < Threshold_k \\ IsC & Otherwise \end{cases} \tag{11}$$

where SC_k represents the set of Similar cases that are situated in the core of cluster k where IsC collects the set of Isolated cases which are more distant to the different clusters centers. We propose to fix $Threshold_k$ as the mean of cases distances towards the cluster ω_k such as:

$$Threshold_k = \frac{\sum_{\boldsymbol{x_i} \in CB; \boldsymbol{x_i} \notin NC} BMD(\boldsymbol{x_i}, \boldsymbol{v_k})}{\#TotalCases - \#NoisyCases} \tag{12}$$

Internal cases detection. For every group of similar cases, we have to flag one as a representative that will cover all of them. We call this case *Internal* since we choose it to be the closest case to the center of each cluster. Formally, we can define Internal cases as follows:

$$\boldsymbol{x_i} \in InC \ \ iff \ \ \exists k; \neg \exists \boldsymbol{x_j} / BMD(\boldsymbol{x_j}, \boldsymbol{v_k}) < BMD(\boldsymbol{x_i}, \boldsymbol{v_k}) \tag{13}$$

where InC represents the set of Internal cases.

4.2.3. Step 3: Case base editing

By reaching this step, the ensemble of all cases are labeled. During the current maintenance step, two types of cases will be removed while the two others will be retained. On the one hand, Noisy cases lead to decrease CBR systems competence in problem-solving. Hence, they should be removed from the CB. For Similar cases, they reduce the CBR system's performance in term of retrieval time without any additional value in term of competence. Therefore, this type of cases, which represents the redundancy, should also be eliminated. On the other hand, Isolated cases are very important towards the overall CB's competence, and their deletion can make some problems permanently unsolvable by the system, so they have to be maintained. Finally, Internal cases should definitely be retained since they cover all the removed similar cases.

4.3. The integrated IMMEP policy for vocabulary and CB maintenance

The IMMEP method provides two-dimensional maintenance for CBR systems. Its principle consists at repeating, for a number of iterations, an alternation between CB learning, on the one hand, and vocabulary learning, on the other hand. From one iteration to another, we make some updates on parameters to conduct the algorithm to the most occur

result. If the first alternation phase concerns features learning, then we apply our strategy of vocabulary maintenance (EVM) as shown in Section 4.1 without proceeding to features maintenance (Step 4). Then, an update of weights, in form of penalization, will be performed on features that are detected, during Step 3, as noises. Further, we mention that we chose to apply iterations of the two alternated phases as times as the number of features. Consequently, we update weights of features flagged as noises, from one iteration to another, as follows:

$$if \;\; f_i \in NF \qquad w_i \longleftarrow w_i - \frac{1}{\#Features} \qquad \forall i = 1..p \tag{14}$$

where w_i is the weight of feature f_i, NF represents the set of features detected as noises in one iteration, and p is the total number of features. At the beginning of every iteration, NF is reinitialized to the empty set ($NF \longleftarrow \emptyset$).

To make use of the updated features weights, we use, for the second phase regarding CB learning, the first step of our new Weighted version of ECTD policy that we call WECM (Section 4.2). This step defines actually a weighted version of ECM algorithm (WECM) that uses a weighted similarity metric. Hence, we apply WECM on the CB using feature weights from phase 1. Then, we detect noisy cases using the credal partition generated by WECM and Equation 8. These cases flagged as noises will be removed before moving to the next iteration. Let note that we keep redundancy in both of cases and features during alternations at the aim of improving learning. Finally, through using outputs from the last iteration, we (1) select only representative features according to Step 4 of vocabulary maintenance (Section 4.1), and (2) remove redundant cases by eliminating those attached to the Similar type (Section 4.2).

For the sake of clarity, we provide, in what follows, an algorithm (Algorithm 1) of our integrated IMMEP maintaining method. After that, we move on to validate our contributions through an experimental study.

Algorithm 1 IMMEP algorithm

Require: Original case base CB with n cases and p features;
K_c: Number of clusters for cases learning;
K_f: Number of clusters for features learning;
/* Some other required parameters are presented during Section 5.1 */
Ensure: Maintained case base CB' with n' cases and p' features (with $n' \leq n$ and $p' \leq p$);

BEGIN
Initialize table of all features weights W to 1.
$CB' \leftarrow CB$
$j \leftarrow 1$ /* *To count the number of iterations* */
while $j < p$ **do**
  /* *Alternate between two phases* */
  Phase 1: Feature learning using cases descriptions
    Apply Steps 1, 2 and 3 of vocabulary maintenance strategy (Section 4.1) using CB' and K_f.
    Set weights for features detected as noisy using Equation 14.
  Phase 2: Case Base learning based on attributes
    Apply Step 1 of WECTD policy for CBM (Section 4.2) using CB', K_c, and W.
    Detect the set of noisy cases NC using Equation 8.
  $CB' \leftarrow CB' \setminus NC$ /* Delete noisiness to improve the next iteration's learning */
  $j \leftarrow j + 1$
end while
Apply Step 4 of vocabulary maintenance strategy (Section 4.1) on CB'.
Apply Steps 2 and 3 of CBM strategy (Section 4.2) on CB'.
END

5. Experimental Analysis

Since the main purpose of the experimentation is to validate the benefit of our contribution in maintaining CBR systems, we devote Section 5.1 to mention the used data and the setting of parameters, Section 5.2 to show the

Table 1: Description of used case bases

Case Base	Ref	Number of attributes	Size	Number of classes	Class distribution
German Credit	GR	2	1000	2	700/300
Sonar	SN	60	208	2	111/67
Breast Cancer	BC	31	569	2	357/212
Phishing	PH	10	1353	3	103/548/702
Glass	GL	9	214	6	70/76/17/13/9/30
Australian	AU	14	690	2	383/307
Indian	IN	10	583	2	416/167
Vehicle	VH	18	946	4	240/240/240/226
Ionosphere	IO	34	351	2	225/126
Yeast	YS	8	1484	10	463/429/244/163/51/44/37/30/20/5

maintenance testing strategy with the evaluation criteria, and Section 5.3 to present three experiments along with their results and discussions.

5.1. Data and parameter settings

The different maintenance methods, in the frame of this paper, are implemented using the software R, version 3.5.2, and tested on 10 case bases from the UCI Machine Learning database repository at California University, Irvine [47]. Each CB from these sets of data presents an ensemble of experiences that are collected from a real-world application in some domain. Their description is given, in Table 1, in term of reference, number of attributes to describe problems, size, and number of classes (or solutions) along with their distribution in dataset.

During the implementation of our Weighted Evidential Clustering and case Types Detection policy for CBM (WECTD), our Evidential Vocabulary Maintenance policy (EVM), and our Integrated Maintaining Method based on Evidential Policies (IMMEP), some parameters have to be fixed especially those for the ECM (Section 2.3) and the k-EVCLUS (Section 2.4) algorithms. Let, on the one hand, set parameters of the ECM algorithm, which serves at regrouping cases, as follows:

- The number of clusters K_c is taken equal to the actual number of solutions for every CB (Table 1).
- The initial prototypes of the different clusters are randomly fixed.
- The exponent α of cardinality, appearing within the cost function J_{ECM} (Equation 4) and the Belief Covariance Matrix Σ (Equation 10), is set equal to 1. It means that, during the optimization process, we do not penalize partitions of clusters with high cardinality.
- The exponent of masses β, appearing in Equation 4, is set to 2, which makes it equal to the exponent of distances within the same function.
- The distance to the empty set partition, denoted δ in Equation 4, is set to 10 (default value)
- The minimum required improvement amount of J_{ECM}, to stop iterations, is set to 0.001.

On the other hand, we set parameters of the k-EVCLUS algorithm, which is used to regroup the set of attributes, such that:

- The number k (referred in k-EVCLUS), which matches the number of distances to compute for each attribute, is taken equal to the number of attributes p (by default). If we set $k < p$, the matrix J of size $p \times k$ is built to contain the indices of attributes where $D[i, j]$ is the distance between attribute i and $J[i, j]$.
- For all the bbms of the initial credal partition, the belief's degrees assigned to all the items of the power set are equitably initialized.
- The parameter d_0 is fixed as the ninth quantile regarding attributes distances matrix D (by default).

- The minimum required improvement amount of J_k, to stop iterations, is set to 10^{-5}.
- The number of clusters K_f, which defines also the retained number of features to express problems' vocabulary, has been varied from 2 to 9. For every fixed value, we measure the accuracy offered by the overall CBR system after applying the corresponding maintaining approach. K_f takes, therefore, the value that corresponds to the highest accuracy for every tested CB through matching between the offered results after applying every implemented vocabulary maintenance policy and the different values of K_f (taken values of K_f are mentioned within Table 3).

Let us mention that some of these tested datasets contain missing values. Hence, we opted to fill this partial knowledge using one of the most known and practiced missingness mechanism called EM imputation, which consists to use the Expectation-Maximization algorithm (EM) [48] to predict incomplete cases regarding missing values at random. Then, we may move on to apply our maintenance strategy for experimentation as shown in the following Section.

5.2. Maintenance testing strategy and evaluation criteria

To evaluate the performance of the different maintenance policies, the following testing strategy, as shown in Figure 5, is performed. Each CB, mentioned in Table 1, has been divided into training set Tr and test set Ts. The maintenance policy is therefore applied on Tr to generate a maintained set of cases named Tr'.

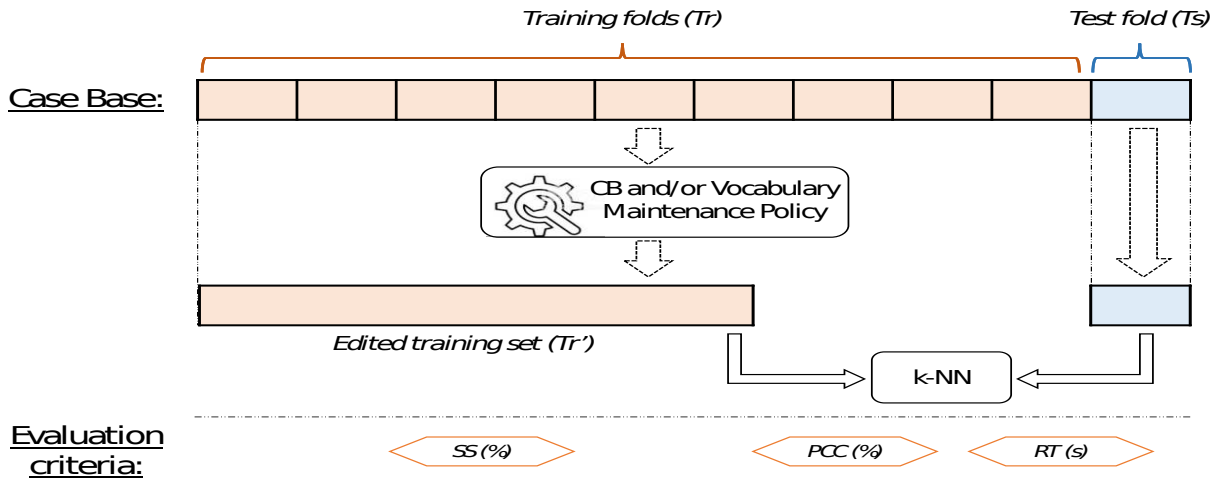

Figure 5: One trial for testing a CBR maintenance policy

Using the k-NN algorithm, which is the most commonly used classification algorithm in CBR systems, the set of problems in Tr will be solved through the labeled experiences in Ts. Therefore, we use the accuracy criterion which is a measure aiming to assess the performance method. It presents the Percentage of Correct Classifications (PCC) and defined such that:

$$Accuracy\ (PCC\%) = \frac{Number\ of\ correct\ classifications\ in\ Ts}{Size\ of\ Ts} \times 100 \tag{15}$$

If we apply different maintenance methods on the same CB with the same k-NN algorithm, then the one that offers the highest accuracy corresponds to a better maintenance performance.
Besides, the time spent (in seconds) by the CBR system to retrieve and classify problems presents our second evaluation criterion, denoted RT for Retrieval Time. Finally, our third evaluation criterion concerns the data retention rate of the maintenance policy, which is defined in term of Storage Size such that:

$$SS(\%) = \frac{Size\ of\ Tr'}{Size\ of\ Tr} \times 100 \tag{16}$$

To obtain the final values of the three evaluation criteria mentioned above (PCC, RT, and SS), we average results of then trials derived from the 10-folds cross validation technique. The idea consists at dividing the CB into ten equivalent folds, and, at each trial, only one fold is used as Ts and the other nine folds are used as Tr. The test set Ts is changed from one trial to another.

5.3. Experimentation and results

The main objective of this paper is to build an integrated editing method for CBR systems that aims to maintain both of the CB and the vocabulary knowledge, which is modeled by the set of attributes. This method presents the integration of two maintenance strategies regarding two levels: WECTD (Section 4.2) for CBM and EVM (Section 4.1) for vocabulary maintenance. That's why, we have performed three experiments in order to show, on the one hand, the performance of CB and vocabulary maintenance strategies separately (*Experiments A* and *B*) and, on the other hand, the performance of their integration, building our IMMEP method, comparing to their simple hybridization (*Experiment C*).

Experiment A. It aims at comparing our case base maintenance strategy (ECTD policy[10]) to two CBM or instance based reduction algorithms, denoted CNN [7] and RNN [8], as well as to the original non-maintained CBR system (Original-CBR). Table 2 shows results offered according to the three evaluation criteria mentioned in Section 5.2.

Table 2: Case base maintenance evaluation

CB	Original-CBR			CNN			RNN			ECTD		
	SS (%)	PCC (%)	RT (s)	SS	PCC	RT	SS	PCC	RT	SS	PCC	RT
GR	100	67.10	0.1018	54.90	54.20	0.0608	54.80	55.25	0.0612	**45.30**	**68.88**	**0.0603**
SN	100	81.28	0.0481	37.55	64.22	**0.0231**	**31.7**	62.85	0.0312	63.54	**84.62**	0.0401
BC	100	59.39	0.0422	62.93	71.45	0.0352	62.93	71.45	0.0355	**55.48**	**74.25**	**0.0337**
PH	100	**87.73**	0.1023	48.75	65.33	0.0718	36.92	60.20	0.0677	**28.49**	84.46	**0.0518**
GL	100	87.38	0.0092	**10.48**	48.33	0.0061	**10.48**	48.33	0.0059	47.82	**89.75**	**0.0051**
AU	100	**64.49**	0.0501	54.06	59.66	0.0328	51.55	59.84	0.0311	**36.11**	64.33	**0.0299**
IN	100	65.26	0.0414	50.08	61.88	**0.0301**	49.84	61.75	0.0314	**37.43**	**67.15**	0.0331
VH	100	58.55	0.0802	64.89	61.24	0.0567	60.45	60.88	0.0478	**50.31**	**61.25**	**0.0412**
IO	100	**86.61**	0.0568	**21.36**	34.46	**0.0297**	**21.36**	34.46	0.0303	44.27	**86.61**	0.0309
YS	100	51.47	0.1125	66.91	50.24	0.0643	66.91	51.02	0.0621	**49.82**	**52.33**	**0.0591**

From the CBM point of view, each one from the three developed approaches (Table 2) offers a strategy for reducing the number of cases, which results to a new storage size comparing to the non-maintained CBR system (Original-CBR) which contains the totality of instances (100%). In term of the SS criterion, our evidential CBM approach (ECTD) has been able to reduce more than half of almost all the tested CBs, where it keeps between about 28 and 63% of the original ones. Comparing to CNN and RNN strategies, competitive results have been offered. For the retrieval time criterion, our ECTD policy offers interesting results, especially comparing to the initial non-maintained CBs. Obviously, reducing CB's size and retrieval time improves the performance of CBR systems. However, we should ensure a high competence which is expressed through the accuracy criterion. In fact, we note, from Table 2, that our ECTD method offers good accuracy values whether comparing to Original-CBR or to the other two case reduction methods. For "Sonar" dataset, for instance, our ECTD method offers an accuracy of 84.62%, where Original-CBR, CNN, and RNN offer accuracy values equal to 81.28%, 64.22%, and 62.85%, respectively.

Experiment B. It is established in order to evaluate the effectiveness of our strategy in maintaining the vocabulary knowledge or the decision making regarding features retention. Our evidential vocabulary maintenance strategy

[10]We are not interested on the elicitation of feature weights if they are not automatically generated. Hence, we use ECTD since it is equivalent to WECTD when setting all weights to 1.

(EVM) has been compared to the Original-CBR and to two well-known feature reduction methods, denoted ReliefF-CBR [49] and InfoGain-CBR [50]. Since, in this experiment, the CB size is unchangeable, results are presented, in Table 3, only in term of accuracy and retrieval time criteria. As already mentioned, the number K_f corresponds to the most convenient number of retained features[11] that is extracted after some tests.

Table 3: Vocabulary maintenance evaluation

CB	Original-CBR		ReliefF-CBR			InfoGain-CBR			EVM		
	PCC (%)	RT (s)	K_f	PCC	RT	K_f	PCC	RT	K_f	PCC	RT
GR	67.10	0.1018	2	71.70	0.0844	3	73.10	**0.0842**	4	**74.11**	0.0845
SN	**81.28**	0.0481		75.42	**0.0321**	9	72.59	0.0419	8	76.07	0.0355
BC	59.39	0.0422	5	**76.66**	**0.0333**	4	75.75	0.0392	5	**76.66**	0.0346
PH	87.73	0.1023	8	86.25	0.0941	7	**88.62**	0.0932	7	**88.62**	**0.0893**
GL	87.38	0.0092	6	89.52	0.0099	6	89.52	0.0091	6	**91.54**	**0.0089**
AU	64.49	0.0501	4	84.90	0.0443	3	84.49	**0.0352**	4	**86.12**	0.0391
IN	65.26	0.0414	8	65.60	0.0301	4	**67.50**	**0.0298**	4	**67.50**	0.0299
VH	58.55	0.0802	4	**69.11**	0.0733	8	61.80	0.0789	6	68.45	**0.0710**
IO	86.61	0.0568	7	90.02	**0.0422**	9	91.44	0.0461	9	**91.45**	0.0466
YS	51.47	0.1125	8	51.47	0.1012	8	51.47	0.0999	7	**56.22**	**0.0998**

From a vocabulary maintenance point of view, our EVM's strategy as well as the implemented ReliefF-CBR and InfoGain-CBR try to select the most powerful attributes in providing the high-competence level of CBR systems, without assigning any interest in maintaining cases instances. Hence, the values of SS criterion, for every method's output, in this experiment, is equal to 100. The most convenient number of selected attributes is referred by K_f, as shown in Table 3. For every CB, the same values of K_f, that offered the best results with EVM approach, have been set during the forthcoming experiment.
Regarding the comparison with Original-CBR, and based on PCC values shown in Table 3, we conclude that our EVM method's strategy has a high maintenance quality. For instance, if a CBR system contains the "Breast Cancer" dataset as a CB, it will know an improvement of competence after applying our EVM policy, going from 59.39% to 76.66%. Comparing to ReliefF-CBR and InfoGain-CBR methods, competitive results are obtained in term of accuracy as well as in term of retrieval time. Nevertheless, our EVM method offers the best results for some CBs such as "Glass" and "Australian" datasets.

Experiment C. After separately evaluating our CBM and vocabulary maintenance strategies, *Experiment C* aims at evaluating the main contribution of the current work regarding our IMMEP method that integrates both of WECTD (Section 4.2) and EVM (Section 4.1) policies. Further, its objective is to compare this integration strategy to the simple hybridization of ECTD and EVM, which consists at (a) applying ECTD then EVM successively (ECTD-EVM), and (b) applying EVM then ECTD successively (EVM-ECTD).

The different results, presented in Table 4, clearly show the significance of our integration strategy embedded within IMMEP method (Algorithm 1) comparing to the two methods regarding the hybridization of the two maintenance levels policies. In term of storage size and retrieval time, our IMMEP method offers an intrinsically better results for almost all the CBs, especially comparing to Original-CBR. Let mention the example of "Phishing" dataset, where the IMMEP method offered a SS equal to 21.85% and a RT equal to 0.0265s, where Original-CB offers values of 100% as storage size, 0.1023s as retrieval time. Comparing to the two other straight hybrid methods, close RT values have been provided, where SS results are to our favor. For example, IMMEP keeps only 28.3% of the original "German" dataset, where ECTD-EVM and EVM-ECTD keep respectively 44.92% and 49.5%.
The two discussed evaluation criteria are obviously important for the performance of any CBR system. However, we should ascertain about their competence, which is defined, in the present work, by the accuracy of provided solutions. In that level, we remark that our new IMMEP method offers the best accuracy for 9 case bases from 10, comparing to

[11] It presents also the number of clusters, during attribute learning, within our EVM and IMMEP methods.

Table 4: Case base and vocabulary maintenance evaluation

CB	Original-CBR			ECTD-EVM			EVM-ECTD			IMMEP		
	SS (%)	PCC (%)	RT (*s*)	SS	PCC	RT	SS	PCC	RT	SS	PCC	RT
GR	100	67.10	0.1018	44.92	65.12	0.0622	49.50	62.45	0.0710	**28.30**	**74.99**	0.0460
SN	100	**81.28**	0.0481	63.54	68.24	0.0219	60.12	59.32	0.0202	**43.22**	81.05	0.0211
BC	100	59.39	0.0422	54.88	44.18	0.0222	**42.13**	51.78	0.0261	56.33	**77.21**	0.0288
PH	100	87.73	0.1023	29.21	75.11	0.0451	39.15	71.04	0.0498	**21.85**	**89.75**	0.0265
GL	100	87.38	0.0092	47.82	79.68	0.0051	48.24	**92.88**	0.0071	**40.46**	**92.88**	0.0077
AU	100	64.49	0.0501	34.82	62.18	0.0315	39.75	59.00	0.0398	**28.75**	**86.06**	0.0254
IN	100	65.26	0.0414	37.44	55.80	0.0271	72.18	63.14	0.0382	**35.53**	**67.15**	0.0253
VH	100	58.55	0.0802	51.36	57.84	0.0503	**50.79**	43.89	0.0511	52.60	**70.12**	0.0601
IO	100	86.61	0.0568	**44.27**	79.76	0.0336	45.25	83.94	0.0344	44.80	**91.56**	0.0299
YS	100	51.47	0.1125	50.75	49.18	0.0621	**32.95**	47.47	0.0589	43.70	**56.21**	0.0607

the straight hybridization of ECTD and EVM (Table 4). Comparing to all the compared methods in experiments *A*, *B*, and *C* (Tables 2, 3, and 4), our main contribution IMMEP shows also a lot of interesting results. For instance, it provided an accuracy of 77.21% for "Breast Cancer" dataset, where Original-CBR, CNN, RNN, ECTD, ReliefF-CBR, InforGain-CBR, EVM, ECTD-EVM, and EVM-ECTD offer, respectively, accuracy values equal to 59.39%, 71.45%, 71.45%, 74.25%, 76.66%, 75.75%, 76.66%, 44.18%, and 51.78%.

As a feedback of the three previous experiments, we can conclude that our proposed strategies have been highly supported by results offered in Tables 2, 3, and 4. Figure 6 presents a snapshot towards the ensemble of the already presented accuracy results in form of 10 bars. The first bar corresponds to results offered by the original non-maintained CBR system, and the others refer to accuracy results offered respectively by three CBM policies, three vocabulary maintenance policies, two hybrid policies, and our proposed integrated maintenance method. Every bar represents the cumulative of PCCs offered by the ten tested case bases (Table 1). Through this accumulation of results offered by the most important criterion in term CB's competence evaluation, we can straightforward note, from Figure 6, the valuable results offered by our contribution of the current work.

These encouraging results have already been discussed during each experiment. However, we note a slightly competence degradation with some case bases after applying some maintenance methods. This fact can be tolerated at the aim of accelerating the time of case indexing and retrieval. The difference between the time criterion values can be more observable if the CBR system uses more complex knowledge type, or some more complex problem solving methods rather than the k-NN. Moreover, in CBR systems, these accuracy values, offered by our strategies of maintenance, may know a notable improvement, going until 100%, if an acceptable level of adaptation effort is applied.

6. Conclusion and Future Work

In this work, we mainly propose an integrated strategy for maintaining CBR systems through removing noisy and redundant cases, as well as irrelevant and redundant features. After providing a try to give a somehow exhaustive state-of-the art of this research field, we presented our IMMEP method, which is based on integrating two dependent strategies for CBR maintenance: The first concerns "Case Base Maintenance", and the second regards "Vocabulary Maintenance". For the sake of efficiency and relevance, we manage knowledge imperfection, that overwhelms such systems, using the belief function theory since it presents one of the most powerful theoretical frameworks for uncertainty management.

Three different experiments have been performed to validate our contributions. Results offered by *Experiment A* supported our case base maintenance in term of cases retention rate, retrieval time, and problem-solving competence. The variation of these offered results is conducted by noisiness and redundancy rate from one case base to another. Through *Experiment B*, we conclude that maintaining cases' vocabulary is also so interesting in term of accuracy. From that point of view, we deduce that it is more important to eliminate noisy features than redundant ones. If we

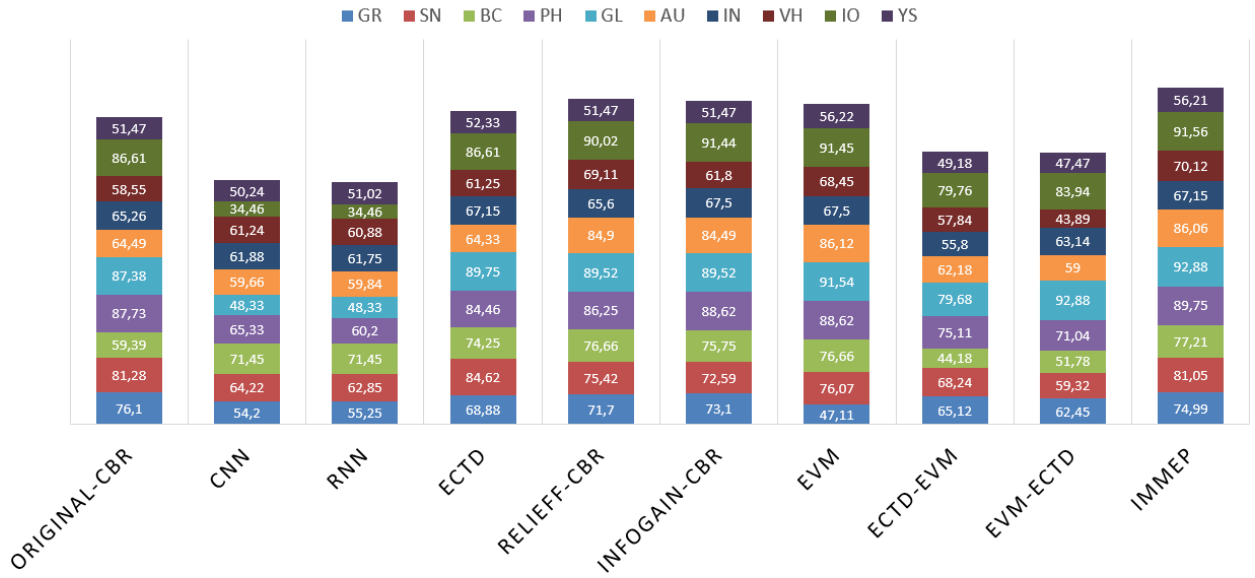

Figure 6: Global accuracy comparison

intend to perform a two-dimensional maintenance task for CBR systems, we can conclude, from *Experiment C*, that it is highly recommended to use our new IMMEP method, that integrates the two latter maintenance strategies, instead of using the straight hybridization of both of them.

Since we focused, in this paper, on maintaining structured CBR systems through their CB and vocabulary editing, we aim, as future work, to tackle the problem of CBR maintenance by giving more interest to the two other knowledge containers, which are similarity measures and adaptation knowledge, while managing uncertainty. Ultimately, more results and comparisons with more recent maintaining approaches will be reported in a forthcoming paper.

References

[1] Richter, M. M., Weber, R. O.: Case-Based Reasoning. *Springer-Verlag Berlin Heidelberg* (2013).

[2] Sene, A., Kamsu-Foguem, B., Rumeau, P.: Telemedicine framework using case-based reasoning with evidences. *Computer methods and programs in biomedicine, 121(1)*, pp. 21-35 (2015).

[3] Sartori, F., Mazzucchelli, A., Di Gregorio, A.: Bankruptcy forecasting using case-based reasoning: The CRePERIE approach. *Expert Systems with Applications, vol. 64*, pp. 400-411 (2016).

[4] Maher, M. L., Pu, P.: Issues and applications of case-based reasoning to design. *Psychology Press* (2014).

[5] Aamodt, A., Plaza, E.: Case-based reasoning: Foundational issues, methodological variations, and system approaches. *In Artificial Intelligence Communications*, pp. 39-52 (1994).

[6] Leake, D. B., Wilson, D. C.: Categorizing case-base maintenance: Dimensions and directions. *In European Workshop on Advances in Case-Based Reasoning*, pp. 196-207 (1998).

[7] Hart, P.: The condensed nearest neighbor rule. *IEEE transactions on information theory, 14(3),* pp. 515-516 (1968).

[8] Gates, G.: The reduced nearest neighbor rule. *IEEE transactions on information theory 18, no. 3,* pp. 431-433 (1972)

[9] Richter, M. M.: The knowledge containers in similarity measures. *Slides of invited talk at the first International Conference of case-based reasoning* (1995)

[10] Ben Ayed, S., Elouedi, Z., Lefevre, E.: Maintaining case knowledge vocabulary using a new Evidential Attribute Clustering method. *In 13th International FLINS Conference on Data Science and Knowledge Engineering for Sensing Decision Support*, pp. 347-354, Springer (2018).

[11] Dempster, A. P.: Upper and lower probabilities induced by a multivalued mapping. *In The annals of mathematical statistics*, pp. 325-339 (1967).

[12] Shafer, G.: A mathematical theory of evidence. *Princeton University Press*, Princeton (1976).

[13] Smets, P.: The transferable belief model for quantified belief representation. *In Quantified Representation of Uncertainty and Imprecision*, pp. 267-301. Springer Netherlands (1998).

[14] Masson, M. H., Denœux, T.: ECM: an evidential version of the fuzzy c-means algorithm. P*attern Recognition 41 (4)*, pp. 1384-1397 (2008).

[15] Denœux, T., Masson, M. H.: EVCLUS: Evidential clustering of proximity data. *IEEE Trans. on Systems, Man and Cybernetics B 34 (1)*, pp. 95-109 (2004).

[16] Denœux, T., Sriboonchitta, S., Kanjanatarakul, O.: Evidential clustering of large dissimilarity data. *Knowledge-based Systems 106*, pp. 179-195 (2016).
[17] Wilke, W., Bergmann, R.: Techniques and knowledge used for adaptation during case-based problem solving. *In Proceedings of the international conference on industrial, engineering and other applications of applied intelligent systems*, pp. 497-506 (1998).
[18] Roth-Berghofer, T. R. and Cassens, J.: Mapping goals and kinds of explanations to the knowledge containers of case-based reasoning systems. *In International Conference on Case-Based Reasoning*, pp. 451-464. Springer (2005).
[19] Richter, M.M.: Knowledge containers. *Readings in Case-Based Reasoning*. Morgan Kaufmann Publishers (2003).
[20] Weber, R.: Fuzzy set theory and uncertainty in case-based reasoning. *Engineering intelligent systems for electrical engineering and communications*, pp. 121-136 (2006).
[21] Lin, S. W., Chen, S.C.: Parameter tuning, feature selection and weight assignment of features for case-based reasoning by artificial immune system. *Applied Soft Computing, 11(8)*, pp. 5042-5052 (2011).
[22] Leake, D., Schack, B.: Flexible feature deletion: compacting case bases by selectively compressing case contents. *In International Conference on Case-Based Reasoning*, pp. 212-227. Springer (2015).
[23] Leake, D., Schack, B.: Adaptation-Guided Feature Deletion: Testing Recoverability to Guide Case Compression. *In International Conference on Case-Based Reasoning*, pp. 234-248. Springer (2016).
[24] Hong, T. P., Liou, Y. L.: Attribute clustering in high dimensional feature spaces. *In International Conference on Machine Learning and Cybernetics, Vol. 4*, pp. 2286-2289. IEEE (2007).
[25] Maji, P.: Fuzzyrough supervised attribute clustering algorithm and classification of microarray data. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 41(1)*, pp.222-233 (2011).
[26] Smyth, B., McKenna, E.: Competence models and the maintenance problem. *Computational Intelligence, 17(2)*, pp. 235-249 (2001).
[27] Smyth, B., McKenna, E.: Modelling the competence of case-bases. *In European Workshop on Advances in Case-Based Reasoning*, pp. 208-220. Springer (1998).
[28] Zadeh, L.A.: Soft computing and fuzzy logic. *In Fuzzy Sets, Fuzzy Logic, And Fuzzy Systems*, pp. 796-804 (1996).
[29] Ben Ayed, S., Elouedi, Z., Lefevre, E.: CEC-Model: A New Competence Model for CBR Systems Based on the Belief Function Theory. *In International Conference on Case-Based Reasoning*, pp. 28-44. Springer (2018).
[30] Brighton, H., Mellish, C.: On the consistency of information filters for lazy learning algorithms. *In Proceedings of european conference on principles of data mining and knowledge discovery*, pp. 283-288 (1999).
[31] Haouchine, M. K., Chebel-Morello, B., Zerhouni, N.: Case Base Maintenance Approach. *In International Conference on Industrial Engineering and Systems Management, IESM'2007, Beijing, Chine* (2007).
[32] Lu, N., Lu, J., Zhang, G., De Mantaras, R. L.: A concept drift-tolerant case-base editing technique. *Artificial Intelligence, 230*, pp. 108-133 (2016).
[33] Mathew, D., Chakraborti, S.: An Optimal Footprint Method for Case-Base Maintenance. *In Florida artificial intelligence research society (FLAIRS) Conference*, pp. 383-388 (2018).
[34] Smiti, A., Elouedi, Z.: Coid: Maintaining case method based on clustering, outliers and internal detection. *In Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, pp. 39-52. Springer (2010).
[35] Sander, F., Ester, M., Kriegelh, P.: The algorithm GDBSCAN and its application. *In Data Mining and Knowledge Discovery*, pp. 178-192 (1998).
[36] Smiti, A., Elouedi, Z.: 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)*, pp. 27-38 (2014).
[37] Ali, R., Ather, M., Ijaz, R., Razzaq, H., Saleem, F., Khan, M. J.: Clustering based deletion policy for case-base maintenance. *In 6th International Conference on Emerging Technologies (ICET)*, pp. 45-48. IEEE (2010).
[38] Smiti, A., Elouedi, Z.: SCBM: soft case base maintenance method based on competence model. *Journal of Computational Science, 25*, pp. 221-227 (2017).
[39] Smiti, A., Elouedi, Z.: Fuzzy density based clustering method: Soft DBSCAN-GM. *In 8th International Conference on Intelligent Systems (IS)*, pp. 443-448. IEEE (2016).
[40] Ben Ayed, S., Elouedi, Z., Lefevre, E.: ECTD: evidential clustering and case types detection for case base maintenance. *In IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA)*, pp. 1462-1469. IEEE (2017).
[41] Ben Ayed, S., Elouedi, Z., Lefevre, E.: DETD: Dynamic Policy for Case Base Maintenance Based on EK-NNclus Algorithm and Case Types Detection. *In International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pp. 370-382. Springer (2018).
[42] Ben Ayed, S., Elouedi, Z., Lefevre, E.: Exploiting Domain-Experts Knowledge Within an Evidential Process for Case Base Maintenance. *In International Conference on Belief Functions*, pp. 22-30. Springer (2018).
[43] Pearson, K.: Mathematical contributions to the theory of evolution. III. Regression, heredity, and panmixia. *In Philosophical Transactions of the Royal Society of London*, pp. 253-318 (1896).
[44] De Amorim, R. C.: A survey on feature weighting based K-Means algorithms. *Journal of Classification, 33(2)*, pp. 210-242 (2016).
[45] Dave, R. N.: Characterization and detection of noise in clustering. *Pattern Recognition Letters*, pp. 657-664 (1992).
[46] Antoine, V., Quost, B., Masson, H. M., Denœux, T.: CECM: Constrained evidential c-means algorithm. *Computational Statistics & Data Analysis*, pp. 894-914 (2012).
[47] Dua, D., Graff, C.: UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. *Irvine, CA: University of California, School of Information and Computer Science* (2019).
[48] Dempster, A. P., Laird, N. M., Rubin, D. B.: Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological), 39(1)*, pp. 1-22 (1977).
[49] Kononenko, I.: Estimating attributes: analysis and extensions of RELIEF. *In European conference on machine learning*, pp. 171-182 (1994).
[50] Lee, C., Lee, G. G.: Information gain and divergence-based feature selection for machine learning-based text categorization. *Information processing & management, 42(1)*, pp. 155-165 (2006).