CIMMEP: constrained integrated method for CBR maintenance based on evidential policies

Safa Ben Ayed, Zied Elouedi, Eric Lefevre

Noname manuscript No.
(will be inserted by the editor)

CIMMEP: Constrained Integrated Method for CBR Maintenance based on Evidential Policies

Safa Ben Ayed · **Zied Elouedi** · **Eric Lefevre**

Received: date / Accepted: date

Abstract The quality of the proposed solutions by Case-Based Reasoning (CBR) systems is highly dependent on recorded experiences and their describing attributes. Hence, to keep them offering accurate and efficient responses for a long time frame, the maintenance of Case Bases (CB) and Vocabulary knowledge is required. However, maintenance operations are usually unable to exploit provided domain-experts knowledge although this kind of systems are widely applied in several real-life contexts. This offered prior knowledge is handled, in our work, in form of pairwise constraints: Regarding cases, Must-Link (ML) affirms that two given problems should have the same solution, and Cannot-Link (CL) informs that two problems cannot have the same solution. These constraints may also regard vocabulary knowledge in such a way that ML is generated when prior knowledge affirm that two given features offer correlated values, therefore, similar information, and CL is built when they provide different information. This paper proposes a new constrained & integrated method, named CIMMEP, encoding *Constrained & Integrated Maintaining Method based on Evidential Policies*, for maintaining both vocabulary and CB through eliminating redundancy and noisiness. Since CBR systems handle real-world experiences, which are full of uncertainty, CIMMEP manages this imperfection using a powerful tool called the belief function theory.

Keywords Case-Based Reasoning · Case Base Maintenance · Vocabulary Maintenance · Constrained Machine Learning · Uncertainty · Belief Function Theory

S. Ben Ayed
Université de Tunis, Institut Supérieur de Gestion de Tunis, LARODEC, Tunis, Tunisia
Univ. Artois, UR 3926, LGI2A, 62400 Béthune, France
Tel.: +33-619548045
E-mail: safa.ben.ayed@hotmail.fr

Z. Elouedi
Université de Tunis, Institut Supérieur de Gestion de Tunis, LARODEC, Tunis, Tunisia

E. Lefevre
Univ. Artois, UR 3926, LGI2A, 62400 Béthune, France

1 Introduction

Case-Based Reasoning (CBR) is an artificial intelligence paradigm for problem solving based on recalling previous experiences which are stored within a memory structure called Case Base (CB) and described by some vocabulary knowledge such as the set of attributes. It is mainly based on the hypothesis that *similar problems have similar solutions* with offering the possibility to make some adaptations to solutions in order to perfectly match new problems characterizations. After the revision of every provided solution, a new case will be stored in the CB in order to provide an incremental learning [1]. However, if we take a snapshot of knowledge in some instant, we may note some irrelevant or out-of-date knowledge which will cause a degradation of the competence as well as the performance of the overall CBR system. For that reason, we find, in the literature, several maintenance policies that have been proposed to revise the content of the CB, the vocabulary describing cases, and other types of knowledge. In CBR systems, this knowledge is distributed over four knowledge containers: the case base, the vocabulary, the similarity measure, and the adaptation rules [2]. The CB is the basic element and plays a special role because cases can be entered without understanding them. The vocabulary refers to the first question that arises: *which data structure is used to represent primitive notions, in general, and the set of cases, in particular?* Hence, our emphasis, in this work, turns around the maintenance of this two knowledge containers.

The proposed maintaining policies aim, generally, at removing *(1)* noisy and/or *(2)* redundant knowledge, using some given strategies. The need of such maintenance operation are more and more important through time since CBR learns incrementally and uses past experiences for adapting solutions. Actually, some of these maintenance policies have the ability to manage imperfection embedded within real-world situations which cause ignorance and overlapping data regions during learning. However, these policies suffer from their disability to help their automatic maintenance mechanisms when prior knowledge, such as those provided by domain-experts, are available. We tackle these problems, in this paper, by proposing a new maintaining policy, called CIMMEP for *Constrained & Integrated Maintaining Method based on Evidential Policies*. By managing constraints, CIMMEP revises the content of the most important maintenance targets simultaneously: *(1) the CB*, which presents the set of numerical experiences, and *(2) the vocabulary knowledge*, which has been restricted to the set of features/attributes describing cases.

Our proposal is characterized by an ability to exploit prior knowledge in order to help the learning task. Hence, it makes use of the semi-supervised learning through pairwise must-link (ML) and cannot-link (CL) constraints. Since we intend to learn both cases and features, we use two kinds of constraints for both types as shown in Fig. 1. On the one hand, during CB learning, we use must-link constraints ML_{CB} to specify that two given cases have the same solution where cannot-link constraints (CL_{CB}) inform that two instances of cases cannot belong to the same cluster. On the other hand, during vocabulary learning, must-link constraints (ML_{Voc}) between two features are generated when prior knowledge affirm that they offer the same information, where cannot-link constraints (CL_{Voc}) are created to affirm that they offer different information during learning. Ultimately, we mention that our new CIMMEP

method is based on some powerful managing uncertainty tools offered within the frame of the belief function theory [3,4] so as to manage imperfection within knowledge containers (CB and Vocabulary) since they present the origins of uncertainty in CBR systems [5].

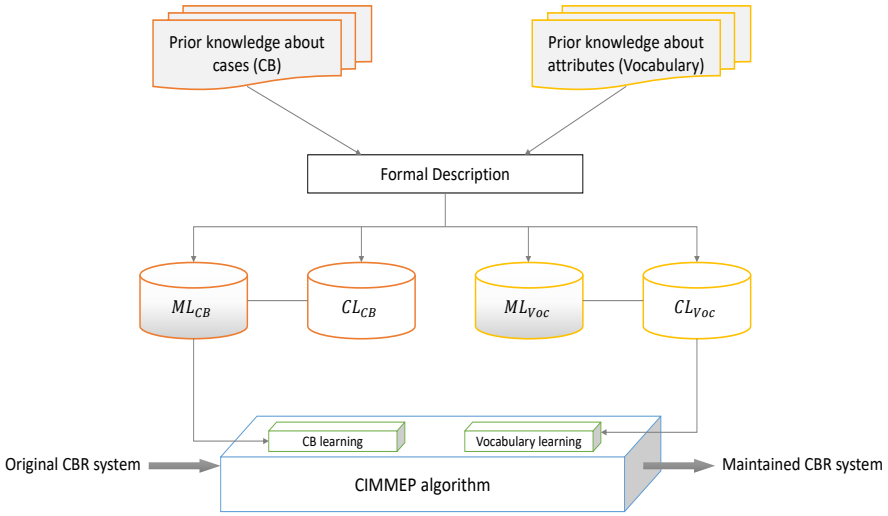

Fig. 1 The used *ML* and *CL* constraints for both CB and Vocabulary maintenance within CIMMEP policy

The rest of this paper is organized as follows. The next Section is dedicated to present some related works regarding policies aiming to maintain CBs, on the one hand, and Vocabulary knowledge, on the other hand. The necessary background related to the belief function theory tools and prior knowledge expression are offered during Section 3. Section 4 is focusing on providing details regarding our proposals for this paper. Throughout Section 5, we present two modes for artificial constraints generation and establish an experimental study followed by results exposition and discussion. Ultimately, conclusion and outlook are stated in Section 6.

2 Case Base and Vocabulary knowledge as maintenance targets

Obviously, CBR systems are designed to operate for a long period of time. However, context variance along with continuous CB learning evolution give rise to the need of maintaining the CB (Section 2.1), on the one hand, and the vocabulary or the set of attributes (Section 2.2), on the other hand. The integration of these both maintenance targets is therefore interesting, especially when offering the possibility to exploit prior knowledge (discussion in Section 2.3).

2.1 Case Base as a maintenance target

The CB presents the basic element for any CBR system which serves to gather the collection of previous experiences to be retrieved. Due to their incremental evolution between-whiles, CBs have to be maintained and cases knowledge should be revised.

By this way, a sub-field of Case Base Maintenance (CBM) arises to revise the organization or content of CBs so as to ease reasoning for a particular set of performance objectives [7]. Hence, CBM has been explored in-depth where we find several policies that aim to maintain CBs, and others to evaluate their maintenance task adequacy degree [8]. An overview about the evolution of this field in term of research work, during the last five decades, is offered in Fig. 2. Let briefly describe this broad spectrum of policies through classifying them into three classes according to the three following strategies.

Selection based strategy consists to select from a case base only the set of representative cases that are able to cover the remaining cases. It embodies, for instance, the Condensed Nearest Neighbor (CNN) policy [9] which consists in iteratively and randomly select cases to be added within a new CB, and test, during every iteration, if it is able to successfully solve all problems of the original one. The Reduced Nearest Neighbor (RNN) policy [10] aims at reducing the CB size, case by case, while no case from the original CB is misclassified by the reduced one. Other variants and methods, that belong to this strategy, are also proposed [11, 12] .

The ensemble of policies belonging to the *Optimization based strategy* are characterized by maintaining CBs through optimizing some evaluation criteria. For instance, Utility Deletion (UD) policy maintains CBs by estimating Minton's utility [13] measurement. Iterative Case Filtering algorithm (ICF) [14] is based on the competence to make decision about cases deletion. Its principle idea is to delete every case that more case can solve it than can solve itself. Besides, RC-NN [15] consists in combining CNN as a CBM policy with the Relative Coverage (RC) metric so as to quantify the competence of cases. Some works that explore the Competence concept within the field of CBR have also been mentioned in Fig. 2 with *orange color*.

Finally, *Partitioning based strategy* gathers CBM policies that handle the overall CBs in form of small ones. These policies use generally the clustering as a machine learning technique to divide CBs. Case clustering has been extensively applied within the CBM field due to its success in detecting cases to be maintained. For example, Clustering, Outliers, and Internal case Deletion policy (COID) [16] uses a density based clustering algorithm to allow defining and removing irrelevant cases. Besides, Evidential Clustering and case Types Detection (ECTD) for CBM [17] as well as its dynamic (DETD) [18] and constrained (CECTD) [19] versions divide the CB with uncertainty management and classify cases into four main types: Two among them are retained where noisy and redundant case types are removed.

2.2 Vocabulary knowledge as a maintenance target

According to authors in [2], the vocabulary knowledge responses to the question *"Which elements of the data structures are used to present fundamental notions?"*.

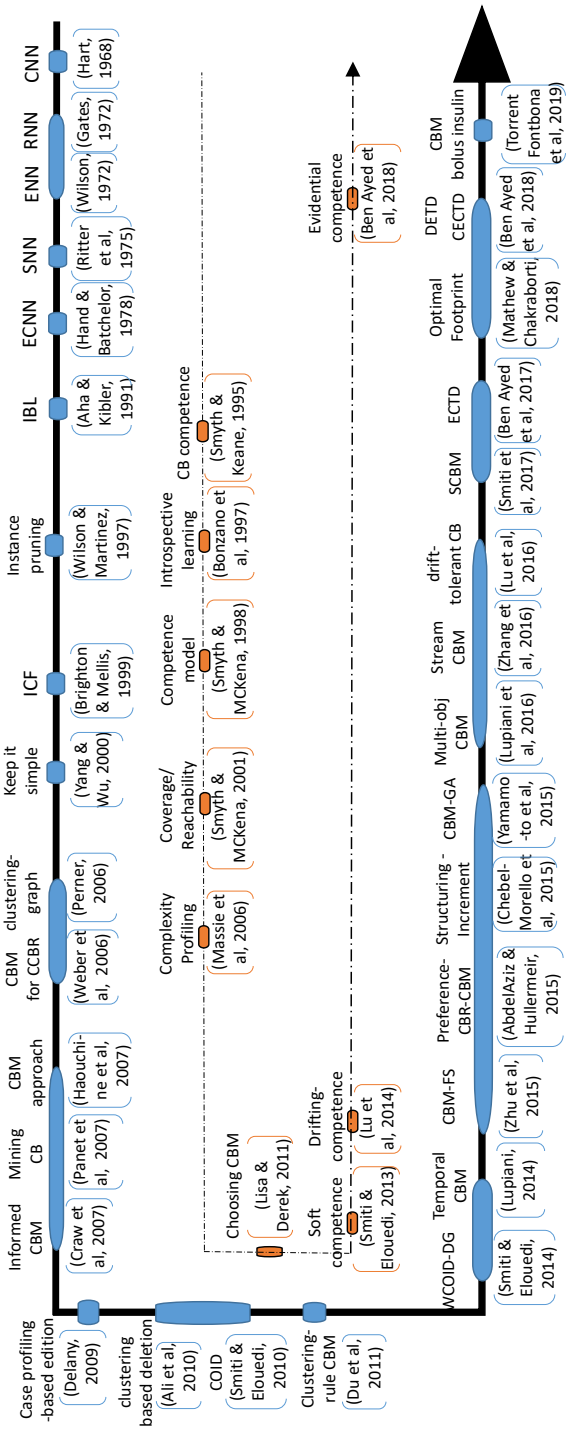

Fig. 2 CBM policies and related concepts over the last five decades

Actually, it is related to the nature of knowledge source within CBR systems. As already mentioned, we restrict this knowledge to be in form of attribute-value data with an object-oriented organization. Obviously, every encountered experience in our real-life may be described with an infinite number of features. However, only some of them are useful to provide the most accurate solution for one problem. As already mentioned, there are basically two types of attributes that should be removed to maintain cases' vocabulary. On the one hand, the set of noisy attributes which their removal from the vocabulary conducts to the improvement of the CBR system's decision making. On the other hand, the set of redundant attributes that we define by the ensemble of high correlated features.

Likewise, we find in the literature various research works aiming to maintain the vocabulary knowledge. An attempt to collect them is provided within Fig. 3.

Actually, the maintenance of vocabulary is widely investigated within Feature Selection (FS) context. By this way, various researches use FS techniques to select/reduce features set for cases description.

Regrouping attributes according to some proximity data has also been proved to be an interesting solution to maintain features within the CBR context. This concept is known by Attribute Clustering (AC), which consists in regrouping similar attributes within the same cluster, where dissimilar attributes are assigned to different ones. In the context of features, similarity reflects generally the relation between them which is generally defined according to the research purpose (e.g., correlation, dependency, etc.). We note that, in the literature, AC has been carried out, for the aim of vocabulary maintenance, in several research works such in [20], [21], and [22].

2.3 Discuss integrated maintenance towards exploiting prior knowledge

For CBM, partitioning CBs and regrouping them into small ones facilitate handling their content since each one may be treated separately. The clustering technique fits such problem since cases may be considered as individuals where distance notion between them is well presented. Concerning vocabulary maintenance, *AC* concept, which consists in regrouping attributes according to some proximity data, is suitable for this problem since it leads to preserve relations between features and offers a high amount of flexibility to the CBR framework, where we can substitute each feature by any other one belonging to the same cluster.

Actually, research established within the two above mentioned sub-fields are somehow interesting. However, we should ask *what if we need to maintain the content of both CB and vocabulary?* Obviously, the naive solution is to successively perform a CBM policy then a vocabulary maintenance policy, or inversely. In fact, in that case, we will even learn on noisy attributes when we maintain CBs, or learn on noisy cases during vocabulary maintenance. Hence, an integration between the two-level maintenance is required. The primary idea of the integrated maintenance approach is to minimize the size of vocabulary (e.g., number of attributes) and the size of the case base simultaneously, which may achieve synergy effect through detecting the most relevant features along with the most representative cases. However, we note that only few studies have been proposed to provide a two-dimensional maintenance.

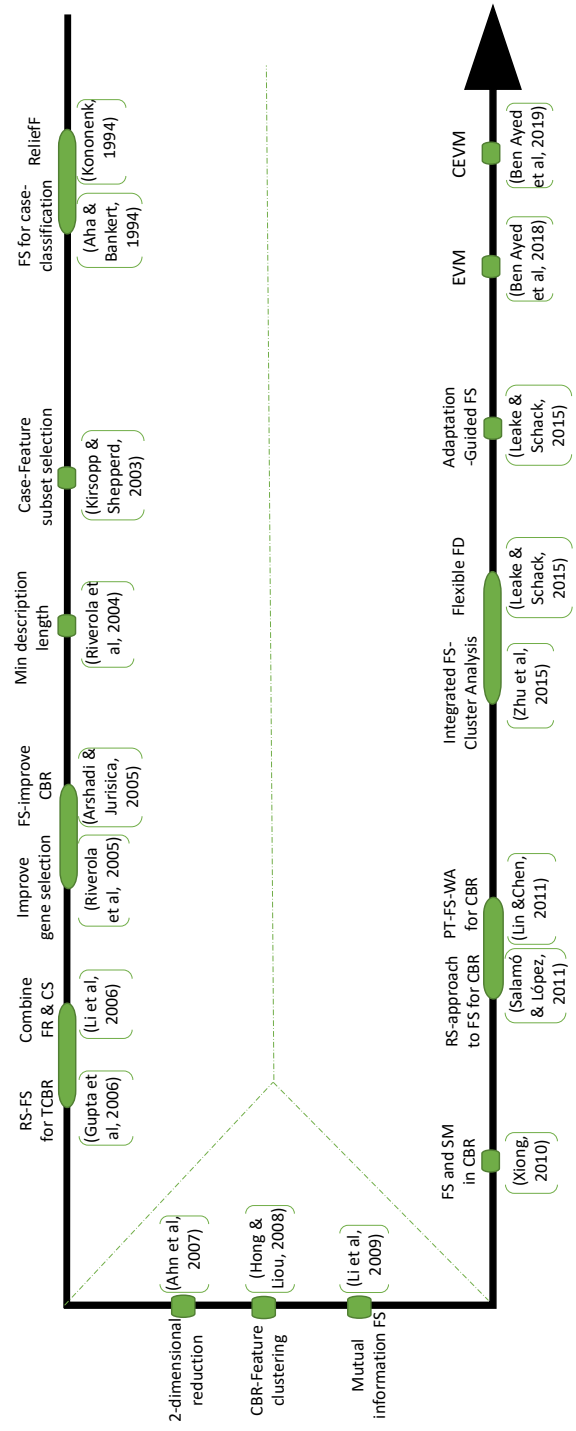

Fig. 3 Related work of vocabulary maintenance within CBR framework

The first one proposed by [23] performs this kind of reduction using the Genetic Algorithm (GA) [24]. Then, authors in [25] proposed similar method that tackles the latter work's limitation towards large data sensibility. Therefore, they employ, for GA search, different models for chromosomes and fitness function. The latter mentioned works have been only applied to artificially produced datasets. To validate a two-dimentional maintenance technique in real-world data, the work, presented in [26], proposes a method for CBR systems with cases and attributes reduction technique for customer classification. However, all these stated works suffer from their disability to manage uncertainty. In fact, we mention that, during every maintenance operation, imperfection within data should be managed within knowledge in general, and within CBR systems, in particular. Vocabulary and CB present two knowledge containers among the origins of uncertainty within CBR systems [5] since they refer to real-world experiences where data is never exact and full by uncertainty, vagueness, and imprecision. Actually, this imperfection could be handled using different theories such as fuzzy logic [27], rough sets [28], or other works on multigranulation [29]. However, in the integrated maintenance context, we find, unfortunately, only one work [30] that uses one among the most powerful tools for this matter called Evidence or belief function theory [3,4]. We aim, in this paper, to address a major limitation of all reviewed methods that figures in their disability to aid learning operations using prior knowledge. In fact, CBR systems are widely applied within various domains where their experts could provide very important knowledge for both CB and vocabulary maintenance tasks. Using pairwise constraints presents one among the most used forms, in the literature, to express provided knowledge. Thus, these extra data, if available, will be used to conduct the most reliable decision instead of being neglected.

Evidential background, as well as the used way to express prior knowledge, are presented in the following Section.

3 Evidential background

To manage uncertainty, our proposals are based on tools offered by the belief function /Evidence/Dempster-Shafer theory [3,4]. Its basic concepts are overviewed in Section 3.1, the evidential clustering along with the credal partition concepts are devoted in Section 3.2, and constraints expression within the evidential framework is presented in Section 3.3.

3.1 Basic concepts of the belief function theory

A belief function model is originally defined by a discrete and finite set of elementary events called the frame of discernment Θ of the problem taken into account. The set 2^{Θ} is called the power set and contains all the possible subsets of Θ. The basic belief assignment (bba) m^{Θ} is a mapping function from 2^{Θ} to $[0,1]$ that assigns to every subset C of Θ a degree of belief reflecting the partial knowledge taken by a variable y defined on Θ, and verifies the constraint $\sum_{C\subseteq\Theta} m^{\Theta}(C) = 1$. For the sake

of simplicity, we use m as notation since we have only one frame on discernment. A bba m is normalized if $m(\emptyset) = 0$. On the opposite case, the interpretation of the mass assigned to the empty set partition consists in measuring the degree of belief towards the hypothesis saying that y does not belong to Θ. This amount of belief can be useful in clustering to identify noises [31]. From a given bba m, the plausibility function is defined, to measure the maximum amount of belief supporting the different subsets in Θ, as follows:

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

Given two bbas m_1 and m_2, defined in the same frame of discernment Θ, the following equation, proposed in [4], presents one of the most known measurements that aim to quantify the degree of conflict between them:

$$\kappa = \sum_{C \cap B = \emptyset} m_1(C)\, m_2(B) \qquad \forall C, B \subseteq \Theta \tag{2}$$

Authors in [32] proved that if two bbas represent evidence regarding two distinct questions and defined in the same frame Θ, then the plausibility that they acquire the same answer is equal to $1 - \kappa$.

3.2 Evidential clustering and credal partition

We call *Evidential Clustering* the task of regrouping objects[1], according to some attribute-based/dissimilarity-based data, within the frame of belief function theory. In an evidential clustering context, the frame of discernment Θ defines the set of a finite number c of clusters. Besides, the uncertainty regarding the membership of an object o_i to the different clusters is modeled by a bba m_i on Θ. If we have n objects, the credal partition is, therefore, the n-tuple composed by n mass functions, such that $M = (m_1, ..., m_n)$ [32]. Generally, M is generated after applying an evidential clustering technique to regroup a set of objects according to their similarity while managing the uncertainty in their membership to all the possible partitions of clusters. Since it quantifies uncertainty in a power set space, the credal partition is more general than hard and soft partitions. Nevertheless, it can be converted to any one of these types [31,32]. After generating the credal partition, the decision about the membership may regard the cluster having the highest pignistic probability, which is defined as follows:

$$BetP(\omega) = \sum_{\omega \in C} \frac{m(C)}{|C|} \qquad \forall \omega \in \Theta \tag{3}$$

[1] In our context, these objects represent the set of features that describe cases, if we handle vocabulary, and the set of cases if we handle CBs.

3.3 Constraints based prior knowledge expression within evidential framework

In real CBR application domains, there is often the case that we possess some extra background knowledge. Actually, it is always gainful to use them throughout learning in general, and during evidential clustering in particular. Within such context, it is beneficial to use instance-level constraints to express background knowledge since they give information about which instances that should or should not belong to the same group. Hence, we use two types of constraints: ML constraints, which indicate that two instances should be associated with the same cluster, and CL constraints which specify that two instances should not belong to the same cluster.

For the search of the credal partition within the belief function framework, these pairwise constraints are translated and expressed as follows. Given m_i and m_j two bbas regarding cluster-membership of instances o_i and o_j respectively, let $pl_{ij}(\Theta_{ij})$ refers to their plausibility to belong to the same cluster, and $pl_{ij}(\overline{\Theta}_{ij})$ refers to the plausibility of the complementary event. They can be calculated as follows [31]:

$$pl_{ij}(\Theta_{ij}) = 1 - \kappa_{ij} \tag{4a}$$

$$pl_{ij}(\overline{\Theta}_{ij}) = 1 - m_i(\emptyset) - m_j(\emptyset) + m_i(\emptyset)\, m_j(\emptyset) - \sum_{k=1}^{c} m_i(\{\omega_k\}) m_j(\{\omega_k\}) \tag{4b}$$

For the sake of clarity regarding the calculation of this plausibility, let mention that it consists in placing ourselves in the Cartesian product $\Theta^2 = \Theta \times \Theta$ and combining the two vacuous extensions of m_i and m_j [32]. If the resulted combination is denoted by m_{ij}, then pl_{ij} can be computed through m_{ij} using Equation 1.

4 Maintaining CB & Vocabulary containers with prior knowledge exploitation

Our purpose behind the current work consists in giving the possibility to exploit prior knowledge during both CB and Vocabulary maintenance in order to repair CBR systems weaknesses. Hence, our proposal defines a constrained and integrated maintenance policy that targets CBR systems knowledge called CIMMEP, for *"Constrained & Integrated Maintaining Method based on Evidential Policies"*. Due to their richness and flexibility, we use tools offered within the belief function theory for knowledge uncertainty management, during the maintenance of CBR systems.

CIMMEP is characterized by an alternation of two main phases so as to provide a trade-off between accurate maintenance tasks for CB and Vocabulary knowledge containers, while exploiting prior knowledge available for both of them. The first phase, which is inspired from our preliminary work described in [33], concerns vocabulary maintenance under constraints, where the second phase which regards CBM under constraints uses steps of a new weighted version of our preliminary work presented in [19]. Therefore, we present, in what follows, our constrained vocabulary maintenance strategy (Section 4.1), our new weighted and constrained policy for CBM (Section 4.2), and finally our main proposal regarding the new constrained and integrated CIMMEP method for both CB and vocabulary maintenance (Section 4.3).

4.1 Constrained Evidential Vocabulary Maintenance (CEVM)

At the aim of performing a high-quality attribute selection within a CBR system, our Constrained Evidential Vocabulary Maintenance policy for CBR systems (CEVM) follows three main steps, as shown in Fig. 4. It consists, first of all, in generating some

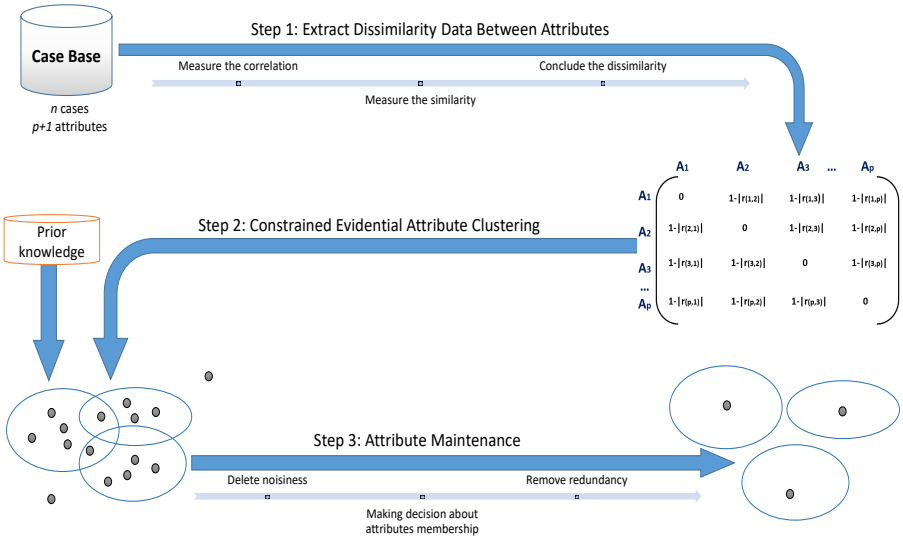

Fig. 4 Steps and substeps of CEVM policy

dissimilarity data, from the CB, between attributes, based on the correlation between their values. Second, CEVM regroups the set of attributes using their dissimilarities and with taking advantage of prior knowledge. After managing uncertainty and generating the credal partition by allowing every attribute to belong to all the partitions of clusters with a degree of belief, we make decision about their membership along with removing noisy and redundant features. More details are given during the three following Subsections.

4.1.1 Step 1: Extracting attributes dissimilarity data

As already mentioned, the notion of dissimilarity between attributes can be defined, according to the context into account, in term of dependency, correlation, etc. In our work, we are interested in identifying the similarity between two features A_1 and A_2 through measuring the correlation between them. This idea comes from the fact that if two features are exactly correlated, then they offer exactly the same information for learning. Let, for instance, refer to Fig. 5 where we show a somehow perfect linear distribution of cases according to both features. We remark, so, that features values

increase with the same amount for A_1 and A_2, which makes them somehow identical or similar.

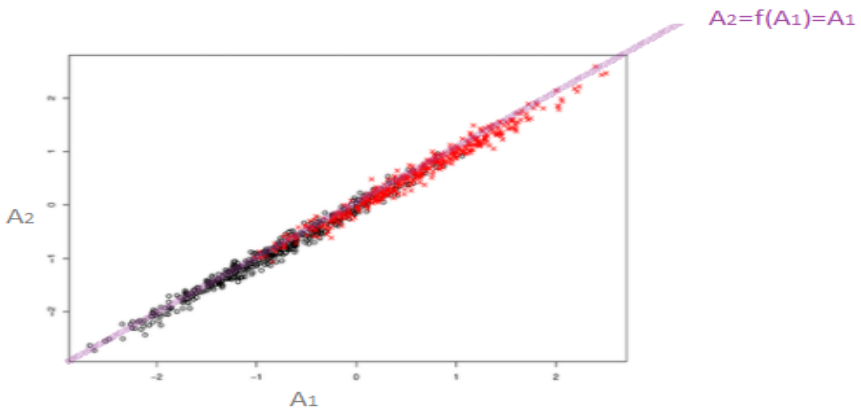

Fig. 5 Example of cases distribution having binary class: A scatterplot of A_1 and A_2

Consequently, to extract attributes dissimilarity data, our CEVM policy follows the three following sub-steps:

1. *Correlation between attributes*: In our context, the origins of dissimilarity data between attributes A_i and A_j are generated through measuring the correlation between their values. The idea is that if two attributes are highly correlated, then they offer the same information for solving problems. We use the well-known Pearson's Correlation Coefficient [34] so as to measure the linear association between the different values a_{il} and a_{jl} of attributes A_i and A_j respectively, as follows:
$$r_{A_iA_j} = \frac{\sum_{l=1}^{n}(a_{il}-\overline{a_i})(a_{jl}-\overline{a_j})}{\sqrt{\sum_{l=1}^{n}(a_{il}-\overline{a_i})^2}\sqrt{\sum_{l=1}^{n}(a_{jl}-\overline{a_j})^2}} \tag{5}$$
where $\overline{a_i}$ and $\overline{a_j}$ are the mean values of features A_i and A_j respectively.
2. *Similarity between attributes*: All correlation values are in $[-1,1]$ [34]. If $r_{A_iA_j} \simeq -1$, then there is a high negative correlation and a high similarity between A_i and A_j since they offer the same information. Similarly, if $r_{A_iA_j} \simeq 1$, then there is a high positive correlation and a high similarity. However, if $r_{A_iA_j} \simeq 0$, then there is no correlation between them, which makes A_i and A_j completely dissimilar. Consequently, we create the square similarity matrix $S = (s_{A_iA_j})$ such as:
$$s_{A_iA_j} = |r_{A_iA_j}| \tag{6}$$
3. *Attributes dissimilarity data*: After measuring the similarity between features, it is straightforward to compute the square dissimilarity matrix $D = (d_{A_iA_j})$ such as:
$$d_{A_iA_j} = 1 - s_{A_iA_j} \tag{7}$$

4.1.2 Step 2: Constrained Evidential Attribute Clustering

During this step, we aim to regroup features according to their similarity and reach the two following objectives during vocabulary maintenance: First, managing the uncertainty in attributes membership to clusters from the complete ignorance to the total certainty using the belief function theory. Secondly, exploiting the prior available knowledge supplied, for instance, by experts of domains in which the CBR is applied. We used a constrained evidential clustering method based on dissimilarity data, which are supplied from the previous step, called Constrained EVidential CLUStering (CEVCLUS) [31]. It is a variant of EVCLUS [32] which is characterized by its ability to take into account a prior knowledge in form of the two pairwise constraints: The Must-link (ML_{voc}) and the Cannot-Link (CL_{voc}) constraints.
To construct the credal partition M, the non-constrained EVCLUS [32] algorithm minimizes a stress function, using a gradient based algorithm, similar to:

$$J(M) = \eta \sum_{i<j} (\kappa_{ij} - \delta_{ij})^2 \tag{8}$$

where $\eta = (\sum_{i<j} \delta_{ij}^2)^{-1}$, and $\delta_{ij} = \varphi(d_{ij})$, with φ is an increasing function such as $\varphi(d) = 1 - exp(-\gamma d^2)$. γ can be calculated as $-log\alpha/d_0^2$, with a recommendation to fix α to 0.05 and d_0, which determines the size of each class, can be set to some quantile of the dissimilarities in D.

The principle of the previous stress function is explained by Equation 4a. It means that if two attributes are too far in term of distance, then they should have a low plausibility to belong to the same cluster, and a large degree of conflict. In our context, if we have prior knowledge informing that attributes A_i and A_j surely belong to different clusters, then the constraints $pl_{ij}(\overline{\Theta}_{ij}) = 1$ and $pl_{ij}(\Theta_{ij}) = 0$ are imposed. In contrast, if prior knowledge affirm that they belong to the same cluster, then the constraints $pl_{ij}(\overline{\Theta}_{ij}) = 0$ and $pl_{ij}(\Theta_{ij}) = 1$ are created. By this way, the CEVCLUS algorithm minimizes, using an iterative gradient-based optimization procedure, the following cost function composed by the sum of EVCLUS's stress function [32] and a penalization term:

$$J_C(M) = J(M) + \frac{\xi}{2(|ML_{voc}| + |CL_{voc}|)} (J_{ML_{voc}} + J_{CL_{voc}}) \tag{9}$$

with

$$J_{ML_{voc}} = \sum_{(i,j) \in ML_{voc}} pl_{ij}(\overline{\Theta}_{ij}) + 1 - pl_{ij}(\Theta_{ij}) \tag{10a}$$

$$J_{CL_{voc}} = \sum_{(i,j) \in CL_{voc}} pl_{ij}(\Theta_{ij}) + 1 - pl_{ij}(\overline{\Theta}_{ij}) \tag{10b}$$

where ML_{voc} (respectively CL_{voc}) presents the set of must-link constraints (respectively cannot-link constraints) about the vocabulary knowledge, and ξ is the hyper-parameter aiming at arbitrating between the stress function and the constraints.

4.1.3 Step 3: Attribute maintenance

Ultimately, we reach the vocabulary maintenance through removing noisy and redundant features, and keeping only those that are unique and represent the different generated clusters during the previous step. As shown in Fig. 4, this step is composed by the three following sub-steps:

1. Removing noisy attributes: Since the performed CEVCLUS clustering method devotes the empty set partition for noisiness allocation, we eliminate attributes characterized by a high belief's assignment to the empty set partition, such that:
$$A_i \in NA \;\; iff \;\; m_i(\emptyset) > \sum_{B_j \subseteq \Theta, B_j \neq \emptyset} m_i(B_j) \tag{11}$$
where *NA* presents the set of noisy attributes.

2. Making decision about attributes membership to clusters through the highest pignistic probability value, using Equation 3.
3. Removing redundancy by keeping only one representative attribute for each cluster. This idea gives an amount of flexibility to the CECTD policy towards CBR framework: If there is a problem in selecting one representative attribute, then we can re-select and re-flag any other attribute from the same cluster.

4.2 Constrained Evidential CB Maintenance (WCECTD): A weighted version

Just before, we presented our strategy for maintain vocabulary knowledge. Now, at the aim of eliminating irrelevant and redundant cases, let show our CBM strategy which performs a new weighted and constrained evidential clustering so as to detect four types of cases and perform maintenance. These three steps, as shown in Fig. 6, are detailed during the following subsections.

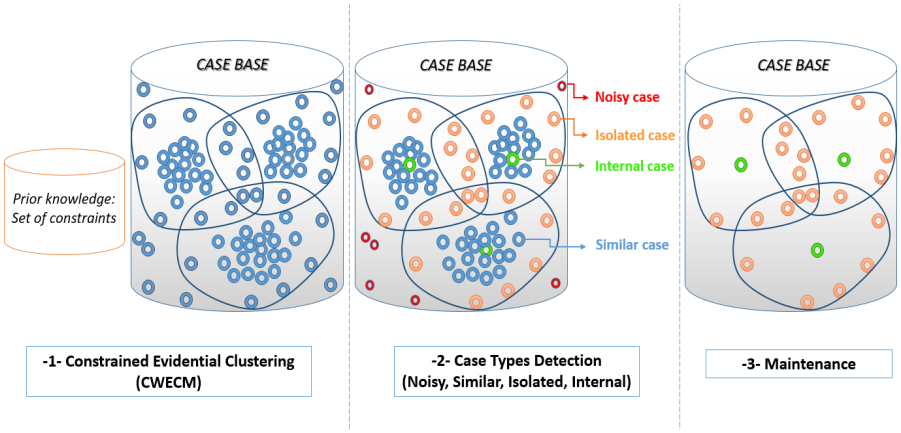

Fig. 6 WCECTD's steps

4.2.1 Step 1: Constrained evidential cases clustering using weighted metric

To learn on case bases with managing uncertainty, exploiting prior knowledge, and taking into account features significance, we propose a new weighted version of the Constrained Evidential C-Means (CECM) [35] clustering technique that we call WCECM. In these methods, noisiness is assigned to the empty set partition. Each object is considered as a case and its class presents its solution part. The background knowledge is presented as case-level constraints regarding CB (ML_{CB} and CL_{CB}).

Objective function and Optimization of the new Weighted version of Constrained Evidential C-Means (WCECM) First of all, let mention that WCECM differs from CECM [35] by its ability to handle weights regarding features importance. Otherwise, they have the same objective function to be optimized in order to build the credal partition. The objective function of the non-constrained version of WCECM (WECM algorithm [30,36]) is defined such that:

$$J_{WECM}(M,V) = \frac{1}{2^c n}\ [\ \sum_{i=1}^{n}\sum_{C_k\neq\emptyset} |C_k|^{\alpha} m_{ik}^{\beta} d_{ik}^2 + \sum_{i=1}^{n} \rho^2 m_{i\emptyset}^{\beta}\ ] \tag{12}$$

subject to:

$$\sum_{j/C_j\subseteq\Omega, C_j\neq\emptyset} m_{ij} + m_{i\emptyset} = 1 \qquad \forall i = 1,..,n \tag{13}$$

where M presents the credal partition of n cases to c clusters, V is the matrix of 2^c clusters centers, d_{ij} represents a Weighted distance between cases $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$, parameters ρ and β serve to treat noisy objects, and α coefficient controls the penalization of subsets of clusters C_j $(j = 1..2^c)$ having high cardinality.

WCECM algorithm is characterized by an additional requirement comparing to WECM: "$pl_{ij}(\theta)$ (respectively $pl_{ij}(\overline{\theta})$) should be as low as possible if $(\boldsymbol{x_i},\boldsymbol{x_j})$ verifies one constraint in CL_{CB} (respectively in ML_{CB})". Hence, its cost function to be minimized is defined as follows:

$$J_{WCECM}(M,V) = (1-\xi)J_{WECM}(M,V) + \xi J_{CONST} \tag{14}$$

where ξ parameter controls the balance between constraints and geometrical model, and J_{CONST}, which indicates CL_{CB} and ML_{CB} violating cost, is defined such that:

$$J_{CONST} = \frac{1}{|ML_{CB}| + |CL_{CB}|}\ [\ \sum_{(\boldsymbol{x_i},\boldsymbol{x_j})\in ML_{CB}} pl_{ij}(\overline{\theta}) + \sum_{(\boldsymbol{x_i},\boldsymbol{x_j})\in CL_{CB}} pl_{ij}(\theta)\ ] \tag{15}$$

For more details about optimizing Equation 14, please see reference [35].

4.2.2 Step 2: Case types detection

We remark that various policies (Fig. 2) opt to divide cases into classes according to their role towards to whole CB competence in solving problems. Our strategy for CBM classifies cases into four types [17], as shown in Fig. 6. *Noisy cases* present a distortion of values, *Similar cases* present a number of cases which are so close to each others (redundant), *Isolated cases* are dissimilar and situated in clusters borders, and *Internal cases* present the center of each group of similar cases (prototypes).

Noisy cases detection: Since CECM algorithm allocates a high belief's degree to the empty set for noisy cases, we propose, as in [17] and [19], to detect them such that:

$$\boldsymbol{x_i} \in NC \;\; iff \;\; m_i(\emptyset) > \sum_{A_j \subseteq \Omega, A_j \neq \emptyset} m_i(A_j) \tag{16}$$

where $\boldsymbol{x_i}$ presents one case and NC presents the set of all the Noisy cases.

Distinguish between Similar and Isolated cases: By arriving to this step, the majority of cases are situated in the core of the different c clusters (Similar cases). However, some cases are isolated and somehow far to their centers (Isolated cases). To make difference between these two types, we compare the distance between cases and clusters' centers to a threshold (Th_k) which is set to the average of cases distances from a given cluster's center. As in [17,19,30], we used the Belief Mahalanobis Distance (BMD) [17] so as to take into account different shapes of distribution, along with managing uncertainty. We make difference, therefore, between Similar and Isolated cases as follows:

$$\boldsymbol{x}_i \in \begin{cases} SC_k & if \; \exists k / BMD(\boldsymbol{x}_i, \boldsymbol{v_k}) < Th_k \\ IsC & Otherwise \end{cases} \tag{17}$$

where SC_k is Similar cases set, IsC is the Isolated cases set, and the threshold Th_k is equal to $\dfrac{\sum_{\boldsymbol{x_i} \notin NC} BMD(\boldsymbol{x_i}, \boldsymbol{v_k})}{\#TotalCases - \#NoisyCases}$.

Flag Internal cases: From each group of Similar cases, we flag an internal case as a prototype or representative to cover all a group of similar cases. Consequently, we detect that case as the closest one to each cluster's center using BMD. Denoting InC as the set of Internal cases, we formally define them as follows:

$$\boldsymbol{x_i} \in InC \;\; iff \;\; \exists k; \neg \exists \boldsymbol{x_j} / BMD(\boldsymbol{x_j}, \boldsymbol{v_k}) < BMD(\boldsymbol{x_i}, \boldsymbol{v_k}) \tag{18}$$

4.2.3 Step 3: Case base maintenance

During cases maintenance, we remove those that are irrelevant or dispensable for the resolution of problems. By this way, we eliminate cases detected as Similar in order to eliminate redundancy and improve response time, along with Noisy cases so as to improve the competence of CBR systems in solving problems.

4.3 The integrated maintaining CIMMEP policy under constraints

Our new constrained and integrated CIMMEP method offers two-dimensional maintenance for CBR systems: Case Base maintenance and vocabulary maintenance. It consists in iterating, for a number of repetitions, a two-phases alternation serving at CB learning, on the one hand, and vocabulary learning, on the other hand. As mentioned during the introduction and in Fig. 1, CIMMEP exploits a formal description arrived from prior knowledge even towards cases or attributes. For every knowledge,

two types of constraints enter as input for our proposal so as to aid learning within its integrated algorithm: ML_{CB} and CL_{CB} are used when we handle cases, where ML_{voc} and CL_{voc} serve to help case attributes learning.

Using these constraints, and after every iteration, we update parameters so as to conduct the algorithm to the most occur output. In case of starting with feature learning phase, our strategy of constrained vocabulary maintenance strategy (CEVM), as shown in Section 4.1, will be applied without performing maintenance step (Step 3). Thus, weights of features that have been flagged as Noisy, will be updated in form of penalization, during Step 3 of WCECTD. Since we choose to apply iterations of the two mentioned phases as times as the number of features, let update weights of features flagged as noises, from one iteration to another, in the following way [30]:

$$if \;\; A_i \in NF \qquad w_i \longleftarrow w_i - \frac{1}{p} \qquad \forall i = 1..p \tag{19}$$

where w_i presents the weight of feature A_i, NF presents noisy features' set regarding one iteration, and the total number of attributes is denoted by p. As initiation of every iteration, NF is defined as an empty set.

Once weights are updated, we perform, during the second phase towards CB learning, Step 1 of our strategy for CBM (Section 4.2) which is able to take into account prior available knowledge as well as features significance rates using weights. In fact, it presents a weighted version of CECM [35] algorithm (WCECM) which utilizes a weighted similarity metric. Consequently, we perform WECM as a machine learning technique on the CB using feature weights from phase 1 and input constraints (ML_{CB} and CL_{CB}). Similarly to CECTD [19], we detect, thus, noisy cases through the generated credal partition and Equation 16, which will be eliminated before proceeding to the next iteration, so as to avoid distorting its operating task. Contrariwise, redundancy in both CB and vocabulary knowledge improves the learning task. Hence, we keep them during the alternation between the two main phases. Ultimately, using results offered within the last iteration, we retain, according to Step 4 of our constrained vocabulary maintenance (Section 4.1), representative attributes only. Besides, we delete all cases considered as redundant so as to improve CBR systems performance. These cases are flagged as *Similar* by our presented CBM strategy (Section 4.2). Let Algorithm 1 details the whole mechanism of our constrained & integrated CIMMEP maintaining method.

Before moving on validating our proposals through an experimental study, we are interested, in this step, to study the computational complexity of our proposal described in Algorithm 1 towards the most important variables: the size of the case base (n), the number of features (p), the number of clusters for cases learning (K_c), and the number of clusters for features learning (K_f). After investigation, we conclude, that it takes polynomial time in the size of the CB (n) and the number of features (p). However, our method takes time exponential in the number of clusters regarding even case learning (K_c) or features learning (K_f), since it handles cases/features' membership not only to clusters but also to all partitions of clusters. This is actually benefit for managing the different levels of uncertainty, but it could be expensive in term of time complexity when we use a large number of clusters. Hence, we will try, during the

next Section, to choose the smallest and the most effective value of clusters number. The whole complexity of our algorithm is estimated to $O(2^{K_f+K_c}.n.p)$.

Algorithm 1 CIMMEP algorithm

Require: Original case base CB with n cases and p features;
List of pairwise constraints: ML_{voc}, CL_{voc}, ML_{CB}, and CL_{CB};
K_c: Number of clusters for cases learning;
K_f: Number of clusters for features learning;
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: Constrained feature learning using cases descriptions
- Apply Steps 1 and 2 of our constrained vocabulary maintenance strategy (Section 4.1) using CB', K_f, ML_{voc}, and CL_{voc}.
- Set weights for features detected as noisy using Equation 19.
Phase 2: Constrained Case Base learning based on attributes
- Apply Step 1 of WCECTD policy for CBM (Section 4.2) using CB', K_c, W, ML_{voc}, and CL_{voc}.
- Detect the set of noisy cases NC using Equation 16.
$CB' \leftarrow CB' \setminus NC$ /* Delete noisiness to improve the next iteration's learning */
$j \leftarrow j+1$
end while
Apply Step 3 of our constrained 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

Throughout this Section, we highlight, first of all, a list of differences between some CBM and/or vocabulary maintenance policies and our CIMMEP method. Then, we establish our experimentation and validate our proposals by developing two variants for both CEVM (Section 4.1) and WCECTD (Section 4.2) policies which differ by their way in generating artificial constraints[2]. For every evidential policy, we will use the variant offering best results to build our integrated CIMMEP method.

5.1 Comparison study between some maintenance policies

Before moving to our experimental investigation, we highlight in Table 1 a list of some differences between our main contribution CIMMEP and some other maintenance policies with which we will compare our results thereafter. We note from this table that our CIMMEP method, which alternates between constrained learning and maintenance, is able to maintain CB and Vocabulary knowledge while exploiting

[2] Calling domain-experts to generate constraints presents one among our perspectives.

prior available knowledge, not sensible to noisy data, and capable to manage uncertainty under the belief function framework. This comparative study has been made in front of the six following maintenance policies: CNN [9], ECTD [17], CECTD [19], ReliefF [37], EVM [20], and CEVM [33].

Table 1 Differences between some maintenance policies

	CNN	**ECTD**	**CECTD**	**ReliefF**	**EVM**	**CEVM**	**CIMMEP**
Maintaining Case Bases	Yes	Yes	Yes	No	No	No	Yes
Maintaining Vocabulary	No	No	No	Yes	Yes	Yes	Yes
Exploiting Prior Knowledge	No	No	Yes	No	No	Yes	Yes
Noisiness Sensibility	Yes	No	No	No	No	No	No
Uncertainty Management	No	Yes	Yes	No	Yes	Yes	Yes
Strategy based on:	Cases Selection	Cases Partitioning	Cases Partitioning	Scores	Attributes Clustering	Constrained Attribute Clustering	Alternation between constrained learning & maintenance

5.2 Constraints generation strategy

To generate *Must-link* and *Cannot-link* constraints, authors in [19] and [33] propose the two following modes:

- Batch mode: It consists in generating simultaneously a number t of constraints (*Must-link* and *Cannot-link*). For instance, we took t equal to 25% of the total number of attributes, when we handle vocabulary, and 10% of the total number of cases when we handle CB knowledge.
- Alternated mode: It consists in alternating between generating one constraint (*Must-link* or *Cannot-link*) and learning, with storing each one incrementally in $listConst$. Similarly, the number of constraints t is taken equal to $\#attributes \times 25/100$ when we handle vocabulary knowledge, and equal to $\#cases/10$ when we handle CB.

How we generate a constraint? We generate artificially a pairwise constraint by handling the uncertainty offered by the credal partition and the pignistic probability transformation (Equation 3). The idea consists in randomly picking two attributes/cases (A_i, A_j)/(x_i, x_j) and behaving according to the three following situations that may arise:

1. If $\exists$ a cluster $\omega / BetP_i(\omega) > Thresh$ and $BetP_j(\omega) > Thresh$, then generate a *Must-link* constraint between A_i and A_j/ x_i and x_j.

2. If $\forall$ clusters $\omega_k / |BetP_i(\omega_k) - BetP_j(\omega_k)| > Thresh$, then generate a *Cannot-link* constraint between A_i and A_j/ x_i and x_j.
3. Else, go back to randomly picking two attributes/cases.

where $Thresh$ is a threshold that aims to answer to the question: *"From which amount of membership certainty in* $[0,1]$*, we consider that the attributes/cases belong or not to the same cluster?"*.
Since the alternate mode for constraints generation proved higher effectiveness in both [19] and [33], it will be used, during the experimental section, to execute and test our proposed methods.

5.3 Data, evaluation criteria, and experimental settings

The presented proposals, within this paper, have been implemented using *R software* and tested on data sets from U.C.I Repository, which are presented in Table 2 by their references, number of attributes, size, number of classes, and their classes distributions.

Table 2 Description of used case bases

Case Base	Reference	Number of attributes	Size	Number of classes	Class distribution
German Credit	GR	20	1000	2	700/300
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

In order to assess their offered maintenance efficiency, we use the three following evaluation criteria:

– The Percentage of Correct Classifications (*PCC*), which refers to the competence of CBR systems in solving new problems, and defined as follows:

$$PCC(\%) = \frac{\#\,Correct\ classifications}{\#\,Total\ classifications} \times 100 \tag{20}$$

– The Retrieval Time (*RT*) Criterion, which measures the time spent to offer all solutions for the different cases instances. It may refer to the performance of CBR systems.
– The Storage Size (SS) which concerns cases data retention rate regarding our CBM and integrated maintenance strategies. It is measured such that:

$$SS(\%) = \frac{\#Size\ of\ Maintained\ Training\ Set}{\#Size\ of\ Original\ Training\ Set} \times 100 \tag{21}$$

To solve cases' problems, we used the 10-fold cross validation technique with the K-Nearest Neighbors (K-NN) since it presents one among the most used machine learning techniques within the CBR framework.

We mention that some of the tested case bases contain missing values. We opted, therefore, to fill this incompleteness through one of the most common and used incompleteness mechanism called EM imputation, which consists to use the Expectation-Maximization algorithm (EM) [38] to estimate missing data within cases' description.

5.4 Results and discussion

Since our main purpose consists in maintaining both case bases and vocabulary with exploiting prior knowledge by handling constraints, we propose performing three experiments: *Experiments (1)* and *(2)* will provide evaluation results related to our strategies for maintaining CB (CECTD) and vocabulary (CEVM) respectively and separately. CECTD's results (Table 3) are compared to the baseline of instance reduction method called CNN [9] and the non-constrained version named ECTD [17], where those of CEVM (Table 4) are compared to one among the most known feature reduction methods, denoted ReliefF-CBR [37], as well as to the nonconstrained version for vocabulary maintenance called EVM [20]. To show the performance of the proposed integration strategy with the availability of prior knowledge, we use *Experiment (3)* and compare results offered by the integration applied by our new CIMMEP method to those offered by the simple hybridization of both CECTD and CEVM (Table 5). This hybridization consists in even applying CECTD then CEVM successively (CECTD-CEVM), or applying CEVM then CECTD successively (CEVM-CECTD). Obviously, all these maintaining policies will also be compared to the original non maintained CBR system (Original-CBR).

According to evaluation criteria mentioned above, we note that our maintaining strategies, which are able to take advantage of prior knowledge, are supported. Obviously, we tolerate sometimes some accuracy degradation at the aim to boost CBR system's performance by accelerating cases retrieval.

Table 3 Evaluating our constrained strategy for case base maintenance

CB	Original-CBR			CNN			ECTD			CECTD		
	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	45.30	68.88	**0.0603**	**44.98**	**69.01**	0.0661
PH	100	**87.73**	0.1023	48.75	65.33	0.0718	**28.49**	84.46	**0.0518**	48.15	86.12	0.0565
GL	100	87.38	0.0092	**10.48**	48.33	0.0061	47.82	89.75	0.0051	55.92	**90.05**	**0.0045**
AU	100	**64.49**	0.0501	54.06	59.66	0.0328	**36.03**	64.33	**0.0299**	36.11	64.34	0.0301
IN	100	65.26	0.0414	50.08	61.88	0.0301	**37.43**	67.15	0.0331	40.06	**68.33**	**0.0298**
VH	100	58.55	0.0802	64.89	61.24	0.0567	50.31	61.25	0.0412	**40.88**	**68.46**	**0.0379**

From a CBM viewpoint, Table 3 shows that our CECTD strategy was able to reduce more than half CBs size (SS%), which leads to decrease the retrieval time

(RT). These results are more interested when we compare them with those offered before maintenance (Original-CBR). However, when we focus on the accuracy criterion which refers to CBs' competence, we note that, by exploiting constraints offered with alternate mode, the CECTD approach provides the best results for four CBs from six: With Vehicle (VH) dataset, for instance, CECTD reduces 40.88% of the initial CB (100%) which leads to decrease cases retrieval time from 0.0802s to 0.0379s, and improves competence from 58.55% to 68.46%. However, CNN and ECTD, which suffer from their disability to handle constraints, reduce 64.89% and 50.31% of the initial VH dataset size, retrieve cases in 0.0567 and 0.012 seconds, and offer accuracy values equal to 61.24% and 61.25% respectively.

Table 4 Evaluating our constrained strategy for vocabulary maintenance

CB	Original-CBR		ReliefF-CBR			EVM			CEVM		
	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	4	74.11	0.0845	4	**74.98**	**0.0733**
PH	87.73	0.1023	8	86.25	0.0941	7	88.62	0.0893	7	**89.06**	**0.0801**
GL	87.38	0.0092	6	89.52	0.0099	6	**91.54**	0.0089	7	86.34	**0.0081**
AU	64.49	0.0501	4	84.90	0.0443	4	86.12	**0.0391**	5	**88.14**	0.0420
IN	65.26	0.0414	8	65.60	0.0301	4	67.50	**0.0299**	4	**68.66**	0.0301
VH	58.55	0.0802	4	**69.11**	0.0733	6	68.45	0.0710	5	69.06	**0.0417**

From a vocabulary maintenance viewpoint, we establish a comparative study with varying the number of clusters or number of selected features K_f from 3 to 9, where the most convenient value for every method and dataset has been chosen. As shown in Table 4, our constrained vocabulary strategy offers high PCCs comparing to the other policies as well as to the original CBs. For example, it provides for Australian (AU) dataset a PCC value equal to 88.14 %, where Original-CBR, RliefF-CBR, and EVM offer values equal to 64.49 %, 84.90 %, and 86.12 %, respectively.
Actually, some degradation of competence, such for Glass (GL) dataset, may not due to our maintenance strategy but to generated constraints quality.

In term of retrieval time, we note competitive results offered by the three vocabulary maintenance policies, with a slightly higher difference comparing to Original-CBR. For instance, "Vehicle" dataset (18 attributes), moved from RT=0.0802s to RT=0.0417s.

Table 5 Evaluating our constrained and integrated strategy for both case base and vocabulary maintenance

CB	Original-CBR			CECTD-CEVM			CEVM-CECTD			CIMMEP		
	SS (%)	PCC (%)	RT (*s*)	SS	PCC	RT	SS	PCC	RT	SS	PCC	RT
GR	100	67.10	0.1018	42.83	70.01	0.0615	48.66	62.44	0.0703	**33.12**	**75.08**	**0.0488**
PH	100	**87.73**	0.1023	30.33	74.14	0.0470	43.54	72.15	0.0508	**28.28**	87.66	**0.0279**
GL	100	87.38	0.0092	50.01	60.05	0.0088	62.11	58.65	0.0112	**46.08**	**93.00**	**0.0076**
AU	100	64.49	0.0501	38.76	62.17	**0.0333**	38.63	62.42	0.0388	**33.76**	**88.21**	**0.0333**
IN	100	65.26	0.0414	**35.37**	58.43	**0.0298**	67.01	64.26	0.0367	44.56	**67.27**	0.0341
VH	100	58.55	0.0802	49.87	59.88	**0.0501**	53.18	45.10	0.0602	**49.81**	**71.16**	0.0589

To validate our main contribution for this paper, we perform the third experiment which results to values offered, in Table 5, after the maintenance of both CB and vocabulary. Hence, our constrained integrated maintenance policy (CIMMEP) has been compared to the simple two-level maintenance hybridization (CECTD-CEVM and CEVM-CECTD)[3].

In term of storage size and retrieval time, our CIMMEP method provides the best results for all the CBs comparing to Original-CBR. Comparing to the straight hybridization maintenance methods, competitive results are offered. However, we have to mention that the most important criterion consists in preserving or improving CBs competence. Actually, in term of accuracy, CECTD-CEVM and CEVM-CECTD gravely decrease some PCC values such for Glass (GL) dataset, where it moves from 87.38% with original-CBR to 60.05% and 58.65%, respectively. However, after performing CIMMEP method, it became equal to 93%.Besides, we note, from Table 5, that our CIMMEP maintenance strategy (Algorithm 1) offers best accuracies for five datasets among the ensemble of six. Moreover, if we compare results of the three performed experiments (Tables 3, 4, and 5), we easily remark interesting results such for AU dataset, where we provide PCC equal to 88.21% and Original-CBR, CNN, ECTD, CECTD, ReliefF-CBR, EVM, CEVM, CECTD-CEVM and CEVM-CECTD offer accuracies equal to 64.49%, 59.66%, 64.33%, 64.34%, 84.90%, 86.12%, 88.14%, 62.17%, 62.42%, respectively.

6 Conclusion

At the aim of giving the ability to CBR systems maintaining policies to exploit prior available knowledge, which may be given by domain-experts, we proposed, in this paper, a constrained and integrated strategy, called, CIMMEP method, which is able to manage uncertainty using the belief function theory. CIMMEP consists in handling constraints to help the automatic learning of knowledge during maintaining simultaneously CBs and vocabulary knowledge within CBR systems.

After providing a somehow exhaustive state-of-the-art, this paper details, on the one hand, two policies for maintaining the CB and the vocabulary separately, and on the other hand, the integrated CIMMEP policy to maintain them simultaneously. As output, CIMMEP removes noisy and redundant knowledge through applying iterations of two main phases. The first one regards the ”Case Base Maintenance” and the second one concerns the ”Vocabulary Maintenance”. Moreover, two ways for artificially generating constraints have been proposed with uncertainty management. The first follows the mode ”Batch” and the second is presented in ”Alternate” mode. Finally, three experiments have been performed to validate our contributions, which have been highly supported by provided results.

We conclude that our proposed method may be applied for any structural CBR system that uses numerical data to describe features' values of cases. Besides, a somehow sizable CBs are recommended when performing our CIMMEP method in order to accurately estimate the correlation between attributes and effectively detect the different types of cases following the constrained evidential learning of cases.

[3] The same constraints are used for the three compared methods

After noting the significance of *Vocabulary* and *CB* knowledge containers, we aim, in future work, to give an interest to maintain the two other knowledge containers, namely *Similarity* and *Adaptation*. Regarding data imperfection handling, we intend also to manage uncertainty not only towards the membership to clusters, but also towards attributes values. Besides, a deeper experimental study will be reported in a forthcoming paper.

References

1. A. Aamodt and E. Plaza.: Case-based reasoning: Foundational issues, methodological variations, and system approaches. *In Artificial Intelligence Communications*, pp. 39-52 (1994)
2. M. M. Richter.: Knowledge containers. *Ian Waston, editor, Readings in Case-Based Reasoning. Morgan Kaufmann Publishers* (2003)
3. A. P. Dempster.: Upper and lower probabilities induced by a multivalued mapping. *Ann. Math. Stat. 38*, pp. 325-339 (1967)
4. G. Shafer.: A Mathematical Theory of Evidence. *Princeton University Press*, Princeton (1976)
5. R. Weber.: Fuzzy set theory and uncertainty in case-based reasoning. *In Engineering intelligent systems for electrical engineering and communications*, pp.121-136 (2006)
6. D. C. Wilson and D. B. Leake.: Maintaining Case-Based Reasoners: Dimensions and Directions. *In Computational Intelligence*, pp. 196-213 (2001)
7. D. B, Leake and D. C. Wilson.: Categorizing case-base maintenance: Dimensions and directions. *In European Workshop on Advances in Case-Based Reasoning*, pp. 196-207 (1998)
8. S. Ben Ayed, Z. Elouedi, and E. Lefevre.: Toward the evaluation of Case Base Maintenance policies under the belief function theory. *In European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, pp. 113-124 (2019)
9. P. Hart.: The condensed nearest neighbor rule. *In IEEE transactions on information theory, 14(3)*, pp. 515-516 (1968)
10. G. Gates.: The reduced nearest neighbor rule. *In IEEE transactions on information theory, 18(3)*, pp. 431-433 (1972)
11. D. L. Wilson.: Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man, and Cybernetics 3*, pp. 408-421 (1972)
12. G. Ritter, H. Woodruff, S. Lowry, T. Isenhour.: An algorithm for a selective nearest neighbor decision rule (Corresp.). *IEEE Transactions on Information Theory, 21(6)*, pp. 665-669 (1975)
13. S. Minton.: Quantitative results concerning the utility of explanation-based learning. *In Artificial Intelligence 42*, pp. 363-391 (1990)
14. H. Brighton, M. Chris Mellish.: On the consistency of information filters for lazy learning algorithms. *European conference on principles of data mining and knowledge discovery. Springer, Berlin, Heidelberg,* pp. 283-288 (1999)
15. B. Smyth and E. McKenna.: Building Compact Competent Case-Bases. *In Case-based reasoning research and development, Lecture notes in computer science*, pp. 329-342 (1999)
16. A. Smiti and Z. Elouedi.: Coid: Maintaining case method based on clustering, outliers and internal detection. *In Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, pp. 39-52 (2010)
17. S. Ben Ayed, Z. Elouedi, and E. Lefevre. ECTD: Evidential Clustering and case Types Detection for case base maintenance. *In IEEE/ACS International Conference on Computer Systems and Applications*, pp. 1462-1469 (2017)
18. S. Ben Ayed, Z. Elouedi, and E. Lefevre.: 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 (2018)
19. S. Ben Ayed, Z. Elouedi, and E. Lefevre.: Exploiting Domain-Experts Knowledge Within an Evidential Process for Case Base Maintenance. *In International Conference on Belief Functions*, pp. 22-30 (2018)
20. S. Ben Ayed, Z. Elouedi, and E. Lefevre.: Maintaining case knowledge vocabulary using a new Evidential Attribute Clustering method. *In International FLINS Conference on Data Science and Knowledge Engineering for Sensing Decision Support*, pp. 347-354 (2018)

21. T. P. Hong and Y. M. Liou.: Attribute clustering in high dimensional feature spaces. *In International Conference on Machine Learning and Cybernetics, 4*, pp. 2286-2289 (2007)
22. P. Maji.: Fuzzy-rough 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)
23. L. I. Kuncheva and L. C. Jain.: Nearest neighbor classifier: Simultaneous editing and feature selection. *Pattern recognition letters, 20(11-13)*, pp. 1149-1156 (1999)
24. J. H. Holland.: Genetic algorithms. *Scientific american, 267(1)*, pp. 66-73 (1992)
25. A. Rozsypal and M. Kubat.: Selecting representative examples and attributes by a genetic algorithm. *Intelligent Data Analysis, 7(4)*, pp. 291-304 (2003)
26. H. Ahn, K. J. Kim, and I. Han.: A case-based reasoning system with the two-dimensional reduction technique for customer classification. *Expert Systems with Applications, 32(4)*, pp. 1011-1019 (2007)
27. L. A. Zadeh.: Fuzzy sets. Information and control, 8(3), 338-353 (1965)
28. Z. Pawlak, J. Grzymala-Busse, R. Slowinski, and W. Ziarko.: Rough sets. *Communications of the ACM, 38(11)*, pp. 88-95 (1995)
29. Y. Qian, J. Liang, C. Dang.: Incomplete multigranulation rough set. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 40(2)*, 420-431 (2009)
30. S. Ben Ayed, Z. Elouedi, and E. Lefevre.: An evidential integrated method for maintaining case base and vocabulary containers within CBR systems, *Information Sciences*, pp 214-229, Vol. 529, Elsevier (2019)
31. V. Antoine, B. Quost, M. H. Masson and T. Denœux.: CEVCLUS: evidential clustering with instance-level constraints for relational data. *Soft Computing, 18(7)*, pp.1321-1335 (2014)
32. T. Denœux and M. H. Masson.: EVCLUS: evidential clustering of proximity data. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 34(1)*, pp. 95-109 (2004)
33. S. Ben Ayed, Z. Elouedi, and E. Lefevre.: CEVM: Constrained Evidential Vocabulary Maintenance policy for CBR systems. *In International Conference on Industrial, Engineering & Other Applications of Applied Intelligent Systems*, pp. 579-592 (2019)
34. K. Pearson.: Mathematical contributions to the theory of evolution. *In Philosophical Transactions of the Royal Society of London*, pp. 253-318 (1896)
35. V. Antoine, B. Quost, M. Masson, T. Denoeux.: CECM: Constrained evidential C-means algorithm. *In Computational Statistics & Data Analysis*, pp. 894-914, Elsevier (2012)
36. M. H. Masson and T. Denoeux.: ECM: An evidential version of the fuzzy c-means algorithm. *Pattern Recognition 41*, pp. 1384-1397 (2008)
37. I. Kononenko.: Estimating attributes: analysis and extensions of RELIEF. *In European conference on machine learning*, pp. 171-182 (1994)
38. A. P. Dempster, N. M. Laird, D. B. Rubin.: Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological), 39(1)*, pp. 1-22 (1977)