CEVM: Constrained Evidential Vocabulary Maintenance policy for CBR systems

Safa Ben Ayed, Zied Elouedi, Eric Lefevre

CEVM: Constrained Evidential Vocabulary Maintenance policy for CBR systems

Safa Ben Ayed[1,2], Zied Elouedi[1], and Eric Lefevre[2]

[1] Université de Tunis, Institut Supérieur de Gestion de Tunis, LARODEC, Tunis, Tunisia
safa.ben.ayed@hotmail.fr, zied.elouedi@gmx.fr
[2] Univ. Artois, EA 3926, Laboratoire de Génie Informatique et d'Automatique de l'Artois (LGI2A), 62400 Béthune, France
eric.lefevre@univ-artois.fr

Abstract. The maintenance of Case-Based Reasoning (CBR) systems has attracted increasing interest within current research since they proved high-quality results in different real-world domains. This kind of systems stores previous experiences, which are described by a vocabulary (e.g., attributes), incrementally in a case base. Actually, the vocabulary presents one among the most important maintenance targets, since it highly contributes in providing accurate solutions and in improving systems' performance, especially within high-dimensional domains. However, there is no policy, in the literature, that offers the ability to exploit prior knowledge (e.g., given by domain-experts) during the maintenance of features describing cases. In this paper, we propose a flexible policy for the most relevant attribute selection based on the attribute clustering concept. This new policy is able, on the one hand, to manage uncertainty using the belief function theory based tools, and on the other hand, to make use of domain-experts knowledge in form of pairwise constraints: If two attributes offer the same information without any added-value, then a *Must-link* constraint between them is generated. Otherwise, if there is no relation between them and they offer different information, then a *Cannot-link* constraint between them is created.

Keywords: Case-Based Reasoning · Vocabulary Maintenance · Belief Function Theory · Uncertainty · Constrained Attribute Clustering.

1 Introduction

Case-Based Reasoning is a methodology of problem-solving that recalls past experiences to solve new problems. It is mainly based on the hypothesis that *similar problems have similar solutions* with offering the possibility to make some adaptations to the provided solution in order to perfectly match the new problem's characterizations. After the revision of every provided solution, the problem-solution couple is retained as a new case within the Case Base (CB) [1]. Over the last three decades, CBR systems have been more and more utilized and applied in several areas such as medicine [2], finance [3], and ecology [4].

Obviously, this can only indicate its strength, success and adequacy even with weak-understandable domains. Since CBR systems are now commercially used, the need of their maintenance presents a key issue for overtime success. Hence, more and more research focus on maintaining the different knowlege containers of CBR systems. Actually, there are four knowledge containers, as defined in [5], that may be maintained [6] within a CBR system: *(1)* Case Base, *(2)* Adaptation, *(3)* Similarity, and *(4)* Vocabulary. Obviously, the CB is the elementary container of any CBR system, that several research aimed to maintain [7]. However, the vocabulary container also presents one among the most important maintenance targets, since it can seen as the basis of the different CBR's steps to offer solutions. For Structural CBR systems (SCBR), we can restrict the vocabulary knowledge to the set of attributes describing cases. By this way, to maintain vocabulary for CBR systems, we are faced to two main challenges: First, the elimination of redundant attributes in order to improve CBR systems performance, especially within large-scale domains. Second, the removal of noisy attributes so as to help the CBR system to be conducted to the most accurate solution. To tackle these challenges in the best way possible, a crucial need of uncertainty management within CBR systems knowledge arises. In fact, cases stored in every CB involve real-world experiences which are never exact. Hence, they cause ignorance and overlapping data regions during learning. This uncertainty within knowledge is managed only in some research, in the literature, that focus on maintaining CBR systems vocabulary via the automatic analysis of their content. However, these works suffer from their disability to aid their automatic maintaining mechanism when prior knowledge regarding attributes, which can be provided by domain experts, are available. This limitation can be tackled through *semi-supervised* learning of features, more precisely the *semi-supervised clustering*. It consists, in our settings, at using the pairwise *Must-link* and *Cannot-link* constraints on some instances to help the used unsupervised attributes clustering. Since we intend to learn on features, *Must-link* constraint[3] between two features is generated when a prior knowledge affirms that they offer almost the same information. On the contrary, a *Cannot-link* constraint[4] is created when prior knowledge is available to affirm that there is no relation between them. Based on these ideas, we build our new vocabulary maintenance policy named CEVM, for "Constrained Evidential Vocabulary Maintenance policy for CBR systems", which manages uncertainty within the framework of belief function theory [14, 15], and allows the exploitation of experts knowledge, related to attributes relations, using the constrained evidential dissimilarity-based clustering technique called CEVCLUS [16].

The remaining of this paper is organized as follows. Section 2 is dedicated to define the vocabulary as a maintenance target in CBR systems with explaining our motivation. The related background on the belief function theory is presented in Section 3. Throughout Section 4, we detail our new established policy aiming at selecting only the most relevant attributes for cases description. We show

[3] Two attributes are surely belonging to the same cluster.
[4] Two attributes cannot belong to the same cluster.

our experimental study as well as our proposed modes for artificial constraints generation in Section 5. Finally, the Section 6 is dedicated for the conclusion.

2 Vocabulary maintenance for CBR systems

Obviously, CBR systems are made to operate for a long period of time. However, the change of the context along with the incremental learning through experiences give rise to the need of maintaining the vocabulary that describes cases.

2.1 The vocabulary as a knowledge container in a CBR system

The vocabulary knowledge is presented and modeled in [5] as the basis of all the other three knowledge containers. In fact, its definition depends mainly on the knowledge source's nature. In this paper, we focus on attribute-value data. However, in non-structural CBR system, more sophisticated methods may be included within the vocabulary container. For our current purpose, we restrict the vocabulary knowledge container to the set of attributes.

2.2 The vocabulary as a maintenance target

Every encountered experience in our real life can be described with an infinite number of features. However, only some of them are useful to provide the 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 that 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. Actually, we call them redundant since they offer the same information, and the removal of one of them does not affect the whole CBR system's competence in solving new problems, but it may improve its performance in term of response time. Within the same road, some works, such in [8–10], target the vocabulary of CBR systems for maintenance. They are mainly based on selecting the most relevant features, where we cite, for instance, the ReliefF method [11] as one among the baselines of features selection methods. However, existing policies suffer from some weaknesses towards the concepts shown in the following Subsections, where we present our motivation.

2.3 Attribute clustering and uncertainty management during vocabulary maintenance

Regrouping attributes according to some proximity data between them can be reached through the attribute clustering concept [12, 13]. Actually, applying this concept during maintaining vocabulary leads to preserve relations between features and offers a high amount of flexibility to the CBR framework, where we

can substitute every attribute by any other one belonging to the same cluster. Similarly to object clustering, we can consider the set of attributes as the set of objects, and regroup them in such a way that features belong to the same cluster are somehow similar. In contrast, attributes belonging to different clusters are dissimilar. However, uncertainty within attributes clustering has to be managed since attributes-values refer to the description of real-world experiences that are full of uncertainty, vagueness, and imprecision. Further, as mentioned in [18], the vocabulary presents one among the origins of uncertainty in CBR framework. That's why, we make use of one among the most powerful tools for this matter called the belief function theory [14, 15], where its basic concepts are shown in Section 3.

2.4 Exploiting prior knowledge during vocabulary maintenance

Usually, research that are interested on knowledge extraction learn via the automatic analysis of available data content without giving the possibility to domain-experts or available prior knowledge to intervene inside this process. Within Case-Based Reasoning framework, systems are generally solving problems within some specific domain, where its experts may provide knowledge that are expensively extracted by machine learning methods. Consequently, it is greatly useful to aid the automatic maintenance process through the exploitation of prior knowledge in form of Must-Link and Cannot-Link constraints. This can be done, for instance, inside a constrained machine learning technique.

3 Belief function theory

To handle uncertainty during the decision making process, we use the belief function theory [14, 15], called also Dempster-Shafer theory or Evidence theory, which is a powerful mathematical framework used to deal with partial and unreliable information in many fields. We show, during this Section, the fundamental concepts of this theory as well as the evidential clustering and the credal partition concepts.

3.1 Fundamental concepts

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 mass (bbm) m^{Θ} is a mapping function from 2^{Θ} to $[0, 1]$ that assigns to every subset A of Θ a degree of belief reflecting the partial knowledge taken by a variable y defined on Θ, and verifies the constraint $\sum_{A \subseteq \Theta} m(A) = 1$. A mass function m is normalized if $m(\emptyset) = 0$. On the opposite case, the interpretation of the mass assigned to the empty set partition consists at 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 [16]. From a given

mass function m, the plausibility function is defined, to measure the maximum amount of belief supporting the different subsets in Θ, as follows:

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

Given two bbms m_1 and m_2, defined in the same frame of discernment Θ, the following Equation, proposed in [15], presents one among the most known measurements that aim to quantify the degree of conflict between them such that:

$$\kappa = \sum_{A \cap B} m_1(A)\ m_2(B) \tag{2}$$

Authors in [17] proved that if two bbms 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[5], 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 K of clusters. Besides, the uncertainty regarding the membership of an object o_i to the different clusters is modeled by a bbm m_i on Θ. If we have n objects, the credal partition is, therefore, the n-tuple composed by n mass functions, such that $M = (m_1, ..., m_n)$ [17]. 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 [16, 17]. After generating the credal partition, the decision about the membership may regard the cluster having the highest pignistic probability, which is defined as follows:

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

In the case of non normalized mass functions, a preprocessing step of normalization for every bbm should beforehand be applied as follows:

$$m_*(A) = \begin{cases} \dfrac{m(A)}{1 - m(\emptyset)} & if\ A \neq \emptyset \\ 0 & Otherwise \end{cases} \tag{4}$$

After presenting the essential background as well as our motivation, we move on now at detailing our contribution for this paper.

[5] In our context, these objects represent the set of features that describe cases.

4 Maintaining vocabulary through evidential constrained attribute clustering

At the aim of performing a high-quality attribute selection within a CBR system, our new Constrained Evidential Vocabulary Maintenance policy for CBR systems (CEVM) goes through three main steps, as shown in Fig. 1. It consists,

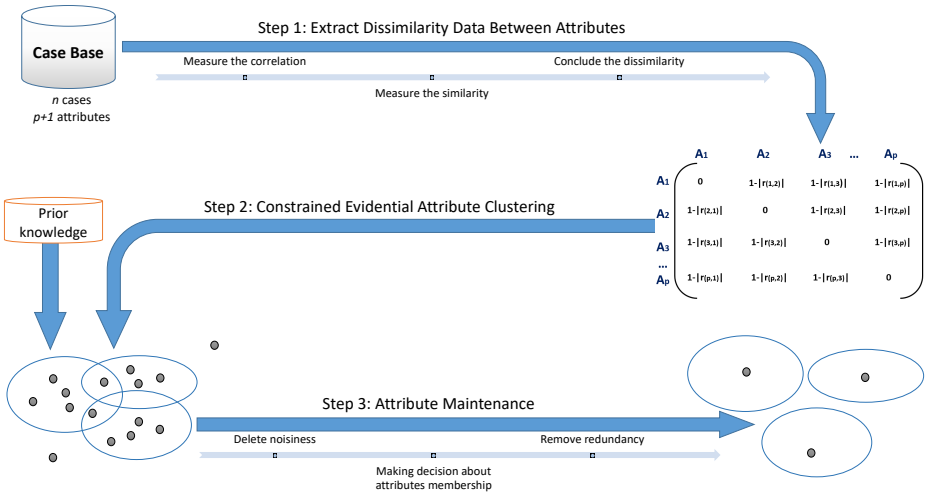

Fig. 1. Steps and substeps of CEVM policy

first of all, at generating some 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 Step 1: Extracting attributes dissimilarity data

The notion of dissimilarity between attributes can be defined, according to the context into account, in term of dependency, correlation, etc. To generate the dissimilarity between attributes, three substeps are followed by our new CEVM policy.

1. *Correlation between attributes*: In our context, the origins of dissimilarity data between attributes 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 Pearson's Correlation Coefficient [19] so as to measure the linear association between the different values a_i and b_i of attributes A and B respectively, as follows:

$$r_{AB} = \frac{\sum_{i=1}^{n}(a_i - \overline{a})(b_i - \overline{b})}{\sqrt{\sum_{i=1}^{n}(a_i - \overline{a})^2}\ \sqrt{\sum_{i=1}^{n}(b_i - \overline{b})^2}} \tag{5}$$

where $\overline{a}$ and $\overline{b}$ are the mean values of features A and B respectively.
The set of correlations between every two features A and B gives rise to a square relational matrix defined as $R = (r_{AB})$.

2. *Similarity between attributes*: All the correlation values in R are bounded between -1 and 1 [19]. If $r_{AB} \simeq -1$, then there is a high negative correlation and a high similarity between A and B since they offer the same information. Similarly, if $r_{AB} \simeq 1$, then there is a high positive correlation and a high similarity between A and B since they offer the same information for learning. However, if $r_{AB} \simeq 0$, then there is no correlation between them, which makes A and B completely dissimilar. As an intuitive consequence, we create the square similarity matrix $S = (s_{AB})$ such as:

$$s_{AB} = |r_{AB}| \tag{6}$$

3. *Attributes dissimilarity data*: After measuring the similarity between features, it is straightforward to compute the square dissimilarity matrix $D = (d_{AB})$ such that:

$$d_{AB} = 1 - s_{AB} \tag{7}$$

By this way, values in D are also in the interval $[0, 1]$.

4.2 Step 2: Constrained Evidential Attribute Clustering

When we have some background knowledge, it is so gainful to use them throughout learning. Actually, this is the main principle of semi-supervised learning. This step, that aims at regrouping features according to their similarity, is very important to reach two other 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 framework. Secondly, exploiting the prior available knowledge supplied, for instance, by the experts of domain in which the CBR is applied. We used a constrained evidential clustering method based on dissimilarity data between objects[6] called Constrained EVidential CLUStering (CEVCLUS) [16]. It is a variant of EVCLUS [17] that is characterized by its ability to taking into account a prior knowledge in form of two pairwise constraints: The Must-link (ML) constraint which concerns two attributes that belong for sure to the same cluster, and the Cannot-Link (CL)

[6] In our policy, it concerns dissimilarity data between attributes, which are supplied from the previous step.

constraint that concerns the pair of attributes that are known to belong to distinct clusters.
Given m_i and m_j two bbms regarding cluster-membership of attributes A_i and A_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 [16]:

$$pl_{ij}(\Theta_{ij}) = 1 - \kappa_{ij} \tag{8a}$$

$$pl_{ij}(\overline{\Theta}_{ij}) = 1 - m_i(\emptyset) - m_j(\emptyset) + m_i(\emptyset)\ m_j(\emptyset) - \sum_{k=1}^{K} m_i(\{\omega_k\})m_j(\{\omega_k\}) \tag{8b}$$

For the sake of clarity regarding the calculation of this plausibility, let mention that it consists at placing ourselves in the Cartesian product $\Theta^2 = \Theta \times \Theta$ and combining the two vacuous extensions of m_i and m_j [17]. If the resulted combination is denoted by m_{ij}, then pl_{ij} can be computed through m_{ij} using Equation 1.

To construct the credal partition M, the non-constrained EVCLUS [17] algorithm minimizes a stress function, using a gradient based algorithm, similar to:

$$J(M) = \eta \sum_{i<j} (\kappa_{ij} - \delta_{ij})^2 \tag{9}$$

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 8a. 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 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 [17] and a penalization term:

$$J_C(M) = stress + \frac{\xi}{2(|ML| + |CL|)}(J_{ML} + J_{CL}), \tag{10}$$

with

$$J_{ML} = \sum_{(i,j)\in ML} pl_{ij}(\overline{\Theta}_{ij}) + 1 - pl_{ij}(\Theta_{ij}), \tag{11a}$$

$$J_{CL} = \sum_{(i,j)\in CL} pl_{ij}(\Theta_{ij}) + 1 - pl_{ij}(\overline{\Theta}_{ij}), \tag{11b}$$

where ξ presents the hyper-parameter aiming at arbitrating between the stress function and the constraints.

4.3 Step 3: Attribute maintenance

Ultimately, we reach our purpose for cases' 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. 1, this step is composed by the three following substeps:

1. Removing noisy attributes: Since the previous applied 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:
$$A_i \in NA \;\; iff \;\; m_i(\emptyset) > \sum_{B_j \subseteq \Theta, B_j \neq \emptyset} m_i(B_j) \tag{12}$$
where A_i represents the attribute i, and NA presents the set of all the 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 our 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.

Example 1. Let consider some CB's vocabulary described by four attributes A_1, A_2, A_3, and A_4. Let us suppose, now, that the frame of discernment contains two clusters ($\Theta = \{cluster_1, cluster_2\}$), and the values of the credal partition $M = [m_1; m_2; m_3; m_4]$ are given by the previous step as shown in Table 1. First, we note, from Table 1, that $m_2(\emptyset)$ is higher than $m_2(\{cluster_1\}) +$

Table 1. Example of credal partition values

M	$\emptyset$	$\{cluster_1\}$	$\{cluster_2\}$	$\{cluster_1, cluster_2\}$
m_1	0.05	0.75	0.15	0.05
m_2	0.65	0.1	0.1	0.15
m_3	0.1	0.05	0.8	0.05
m_4	0.2	0.1	0.5	0.2

$m_2(\{cluster_2\}) + m_2(\{cluster_1, cluster_2\})$. Then, according to Equation 12, we

Table 2. Pignistic probability transformation values

	$cluster_1$	$cluster_2$
$\boldsymbol{BetP_1}$	**0.8158**	**0.1842**
~~$BetP_2$~~	~~0.5~~	~~0.5~~
$\boldsymbol{BetP_3}$	**0.0833**	**0.9167**
$BetP_4$	0.25	0.75

flag A_2 as a noisy attribute ($A_2 \in NA$). Consequently, we update the CB's vocabulary by removing the second attribute A_2. Then, we make decision about attributes membership to clusters using BetP defined in Equation 3. Their corresponding pignistic probability values are shown in Table 2, from which we can conclude that A_1 belongs to $cluster_1$, and A_3 and A_4 belong to $cluster_2$. Finally, we keep only the attribute A_1 as representative of $cluster_1$ and the attribute A_3 as representative of $cluster_2$ to describe the new maintained case base vocabulary.

5 Experimental analysis using artificial constraints

Throughout this Section, we establish our experimentation and validate our contribution by developing two variants of our CEVM policy that differ by their way in generating artificial constraints[7].

5.1 Constraints generation strategy

Two main modes for *Must-link* and *Cannot-link* constraints generation, such in [7], are build during our experimental analysis:

– Batch mode for constraints generation (CEVM$_{bat}$): It consists at 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. We store the list of these constraints in $listConst$. The activity diagram of CEVM$_{bat}$ is shown in Fig. 2.

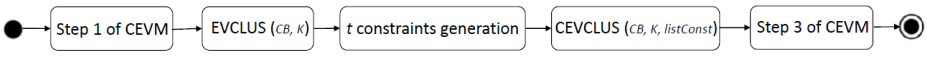

Fig. 2. Activity diagram for batch mode constraints generation

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

– Alternated mode for constraints generation (CEVM$_{alt}$): It consists at 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 $\#attributes \times 25/100$. Its activity diagram is shown in Fig. 3.

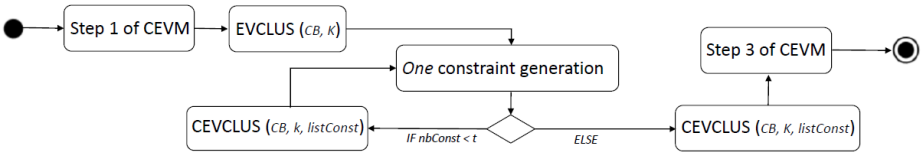

Fig. 3. Activity diagram for alternate mode constraints generation

How we generate a constraint? Actually, 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 at randomly picking two attributes A_i and A_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 the attributes A_i and A_j.
2. If $\forall$ clusters $\omega_k / |BetP_i(\omega_k) - BetP_j(\omega_k)| > Thresh$, then generate a *Cannot-link* constraint between the attributes A_i and A_j.
3. Else, go back to randomly picking two attributes.

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* A_i *and* A_j *belong or not to the same cluster?"*[8].

5.2 Data, evaluation criteria, and experimental settings

Our new CEVM policy has been implemented using *Matlab R2015a* and the default values for the different CEVCLUS [16] method's parameters have been taken. CEVM with its two variants have been tested on six data sets from U.C.I Repository[9] where the set of attributes are considered as the vocabulary describing cases' problems, and their classes refer to cases' solutions.

In order to assess the efficiency of our new vocabulary maintenance policy, we use the two following evaluation criteria:

[8] During the experimentation, different values to set $Thresh$ have been tested. The best results are offered with $Thresh = 0.55$.

[9] Sonar (SN), Ionosphere (IO), Glass (GL), BreastCancer (BC), German (GR), and Heart (HR): https://archive.ics.uci.edu/ml/

– The Percentage of Correct Classifications (PCC), which is defined as follows:

$$PCC(\%) = \frac{\#\ Correct\ classifications}{\#\ Total\ classifications} \times 100 \quad (13)$$

The PCC criterion refers to the competence of CBR systems in solving new problems.
– The Retrieval Time (RT) Criterion, which measures the time spent to offer all the solutions for the different cases instances. It may refer to the performance of CBR systems.

To solve cases' problems, we use the K-Nearest Neighbor (K-NN) since it presents one among the most used machine learning techniques within the CBR framework. We choose to apply *3-NN* so as to avoid the effect of noisy cases during learning. Hence, the RT criterion is exerted around that *3-NN* method. To offer the final results towards the PCC, the 10-fold cross validation technique is used.

5.3 Results and discussion

In our comparative study, as shown in Table 3, we present results offered by five different sources where two among them present the two variants of our contribution for this paper (CEVM$_{bat}$ and CEVM$_{alt}$), and the three others present the non maintained case base (Original-CBR), the reliefF method [11] for feature selection (ReliefF-CBR), and the non-constrained vocabulary maintenance policy EvAttClus [8]. Results in Table 3 are offered after varying the number of clusters, or the number of the selected attributes K, from 3 to 9. The most convenient K for every method and every data set is chosen[10].

Table 3. Accuracy and retrieval time evaluation

	CB	Original-CBR		ReliefF-CBR		AttEvClus		CEVM$_{bat}$		CEVM$_{alt}$	
		PCC(%)	RT(s)	PCC(%)	RT(s)	PCC(%)	RT(s)	PCC(%)	RT(s)	PCC(%)	RT(s)
1	SN	73.07	0.0642	71.15	0.0078	74.51	0.0082	74.51	0.0081	**75.12**	0.0079
2	IO	86.04	0.0223	84.90	0.0121	**88.03**	0.0085	**88.03**	0.0082	**88.03**	0.0087
3	GL	88.79	0.0141	87.38	0.0089	87.98	0.0093	88.79	0.0092	**91.34**	0.0081
4	BC	96.04	0.0199	96.45	0.0097	**96.63**	0.0097	**96.63**	0.0098	**96.63**	0.0099
5	GR	70.60	0.0319	69.60	0.0119	71.21	0.0122	71.21	0.0123	**73.25**	0.0121
6	HR	56.80	0.0276	59.86	0.0089	60.88	0.0081	**62.78**	0.0078	**62.78**	0.0071

In term of accuracy, both of our two variants CEVM$_{bat}$ and CEVM$_{alt}$ offer good results comparing to the other methods as well as to the original non-maintained case bases (Original-CBR). We note that the alternate mode for

[10] ReliefF-CBR: GL and GR ($K = 7$); BC ($K = 8$); SN, IO, and HR ($K = 9$);
EvAttClus: IO ($K = 3$); HR ($K = 5$); BC and GR ($K = 8$); SN and GL ($K = 9$);
CEVM$_{bat}$: IO ($K = 3$); HR ($K = 4$); BC and GR ($K = 8$); GL and SN ($K = 9$);
CEVM$_{alt}$: IO ($K = 3$); HR ($K = 4$); GR ($K = 6$); GL ($K = 7$); SN and BC ($K = 8$);

constraints generation is more efficient than the batch mode. We can conclude, furthermore, that better results may be offered if we resort to domain experts. In our context, both of CEVM's variants, that are able to make use of prior knowledge in form of constraints, have been able to maintain all the tested CBs' vocabulary with preserving or even improving their competence in solving problems. For example, they offer PCCs equal to 88.79% and 91.34% for "Glass" data set, where ReliefF-CBR and AttEvClus methods offer PCCs equal to 87.38% and 87.98% respectively. These results can only be explained by CEVM strategy's efficiency in detecting noisy and redundant features. In term of retrieval time, we note very competitive results offered by the four vocabulary maintaining policies. However, a slightly higher difference in RT are noted towards Original-CBR. For instance, "Sonar" data set (60 attributes), moved from RT=0.0642 s to RT=0.0079 s with CEVM$_{alt}$.

6 Conclusion

In this paper, a new vocabulary maintenance method for CBR systems, called CEVM, with two modes for artificial constraints generation (batch and alternate mode), are proposed. In order to aid its automatic maintaining process, the proposed policies CEVM$_{bat}$ and CEVM$_{alt}$ offer an ability to exploit prior knowledge in form of pairwise constraints within a constrained clustering method. They are also able to manage the uncertainty thanks to the framework of the belief function theory. Finally, the attribute clustering concept for feature selection makes our new CEVM method more flexible for maintaining vocabulary within CBR framework. During experimentation, better results are offered by CEVM$_{alt}$ version than by CEVM$_{bat}$.

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. D. Glez-Pea, F. Daz, J. Hernndez, J. Corchado, and F. Fdez-Riverola. geneCBR: a translational tool for multiple-microarray analysis and integrative information retrieval for aiding diagnosis in cancer research. *BMC Bioinformatics 10:187*, 2009.
3. C. L. Chuang. Application of hybrid case-based reasoning for enhanced performance in bankruptcy prediction. *Information Sciences, 236*, pp. 174-185, 2013.
4. A. Lesniak, K. Zima. Cost calculation of construction projects including sustainability factors using the Case Based Reasoning (CBR) method. *Sustainability, 10(5), 1608*, 2018.
5. M. M. Richter and M. Michael. Knowledge containers. *In Readings in Case-Based Reasoning (Morgan Kaufmann)*, 2003.
6. D. C. Wilson and D. B. Leake. Maintaining Case-Based Reasoners: Dimensions and Directions. *In Computational Intelligence*, pp. 196-213, 2001.
7. 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, Springer, Cham, 2018.

8. S. Ben Ayed, Z. Elouedi, and E. Lefevre. 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.
9. N. Arshadi and I. Jurisica. Feature Selection for Improving Case-Based Classifiers on High-Dimensional Data Sets. *In Florida Artificial Intelligence Research Society Conference (FLAIRS)*, pp. 99-104, 2005.
10. D. Leake and B. Schack. Flexible feature deletion: compacting case bases by selectively compressing case contents. *In International Conference on Case-Based Reasoning*, pp. 212-227, Springer, Cham, 2015.
11. I. Kononenko, Estimating attributes: analysis and extensions of RELIEF. *In European conference on machine learning*, pp. 171-182, Springer, 1994.
12. T. P. Hong and Y. L. Liou. Attribute clustering in high dimensional feature spaces. *In International Conference on Machine Learning and Cybernetics, Vol. 4*, pp. 2286-2289, IEEE, 2007.
13. P. Maji, Fuzzy-rough supervised attribute clustering algorithm and classification of microarray data. *In Transactions on Systems, Man, and Cybernetics,Part B (Cybernetics)*, pp. 222-233, IEEE, 2011.
14. A. P. Dempster. Upper and lower probabilities induced by a multivalued mapping. *Ann. Math. Stat. 38*, pp. 325-339, 1967.
15. G. Shafer. A Mathematical Theory of Evidence. *Princeton University Press*, Princeton, 1976.
16. 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.
17. 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.
18. R. Weber. Fuzzy set theory and uncertainty in case-based reasoning. *Engineering intelligent systems for electrical engineering and communications*, pp.121-136, 2006.
19. K. Pearson, Mathematical contributions to the theory of evolution. *In Philosophical Transactions of the Royal Society of London*, pp. 253-318, 1896.