DETD: Dynamic policy for case base maintenance based on EK-NNclus algorithm and case Types Detection

Safa Ben Ayed, Zied Elouedi, Eric Lefevre

DETD: Dynamic policy for case base maintenance based on EK-NNclus algorithm and case Types Detection

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

[1] LARODEC, Institut Supérieur de Gestion de Tunis, Université de Tunis, Tunisie,
safa.ben.ayed@hotmail.fr, zied.elouedi@gmx.fr
[2] Univ. Artois, EA 3926, LGI2A, 62400 Béthune, France,
eric.lefevre@univ-artois.fr

Abstract. Case Based Reasoning (CBR) systems know a success in various domains. Consequently, we find several works focusing on Case Base Maintenance (CBM) that aim to preserve CBR systems performance. Thus, CBM tools are generally offering techniques to select only the most potential cases for problem-solving. However, cases are full of imperfection since they represent real world situations, which makes this task harder. In addition, new problems having substantially new solutions will be found in case bases over the time. Hence, we aim, in this paper, to propose a new CBM approach having the ability to manage uncertainty and the dynamic aspect of maintenance using the evidential clustering technique called EK-NNclus based on belief function theory, where clusters' number is fixed automatically and changes from one maintenance application to another. Finally, the maintenance task is performed through selecting only two types of cases.

Keywords: case based reasoning, case base maintenance, belief function theory, uncertainty, clustering, EK-NNclus

1 Introduction

Case Based Reasoning is an analogy-based problem-solving paradigm which learns from old experiences using a memory called case base [1]. The strength of CBR systems can be summed up through their ability to offer a high-quality solution even with a weak-understanding domains. Besides, CBR systems are characterized by an incremental learning since each solved problem will be stored in order to serve for future problems resolution. There is a wide range of successful CBR applications in several domains such that diagnosis [2], design [3], help desk [4], decision support [5], etc. Like all other applications that are designed to work over long period of time, CBR systems need a maintenance task, especially regarding their case bases, since their quality presents the key success of all the system. Actually, the case base of a CBR system should contain only relevant cases in order to improve its competence on problems resolution on the one hand, and its performance by reducing the research time on the other hand. For this reason, case base maintenance operations are generally opting to select from the case base the most competent ones. To obtain that, a number of aspects should be taken into account while maintenance.

First, the maintenance task should take into account the uncertainty aspect. In fact, within real world situations, information are often imprecise, uncertain and/or incomplete. Hence, the most powerful cases in problem resolution cannot be well defined without considering the uncertainty aspect that can be caused by unquantifiable data, user ignorance and/or overlapping of data regions.

Second, CBR systems are exposed with time and with users requirement evolution to new types of solutions for new problems. Hence, the dynamicity in solutions should be managed while the maintenance of case bases which reflects modern and contemporary environment. To manage these solutions along with their problems in the case base, some CBM approaches [10] [11] [12] opt to partition cases using a clustering technique as a preprocessing task so as to learn from the case base and devise it into a number of small ones. Then, they select the most representative cases from each cluster.

However, the dynamic aspect consists, in our context, on the capacity of the CBM approach to fix dynamically and automatically the number of clusters regardless which time we perform the maintenance. Besides, existing CBM approaches are not offering a dynamic maintenance because they are suitable only for static collections of cases, and their offered maintenance should be accompanied every time by prior information to be well re-applicable. Actually, and to the best of our knowledge, the dynamicity in case bases maintenance for CBR systems is the most neglected aspect in CBM field.

To manage these two aspects while maintenance, we propose, in this paper, a new approach for case base maintenance that uses the belief function theory [8][9] as one of the most powerful tools for handling uncertainty, more accurately a dynamic evidential machine learning technique called EK-NNclus [7].

The rest of this paper is organized as follows. In Section 2, we review some of CBM methods based on clustering technique. The necessary background regarding the belief function theory and the used evidential clustering technique called EK-NNclus [7] are offered in Section 3. Section 4 describes the different steps of our proposed approach called **DETD** for "**D**ynamic policy for case base maintenance based on **E**K-NNclus algorithm and case **T**ypes **D**etection". Finally, the experimentation is shown during Section 5.

2 Case Base Maintenance: Partitioning based policies

Case Base Maintenance represents a fundamental task aiming to give CBR systems the ability to solve effectively new problems within a reasonable time since they are faced to a large number of cases with a continuous evolution [6]. Conspicuously, the most intuitively way to deal with large case bases while maintenance is to divide them into a number of small ones. Consequently, it will be easier to handle them. Nevertheless, we find in the literature several policies belonging to other strategies as shown in [10]. In the remaining of this Section, we review some CBM policies that use the partition strategy, more accurately, the clustering as a machine learning technique.

On the one hand, with considering partition strategy, we find in the literature hard CBM policies that are not able to deal with uncertainty in data. For instance, we cite COID method [11] encoding *Clustering, Outliers and Internal case Deletion* which is based on a density-based clustering technique for cases gathering and noisy cases detection. Then, it computes cases-clusters distances to flag outliers and internal cases so as to perform the maintenance. As an extention of COID, we find among others, WCOID method [12] which appends a feature weighting technique to give more importance to the most "informative" features in term of problem solving while the maintenance. However, this type of policies is generally reducing the case bases competence since they suffer from their disability to manage uncertainty in cases involving real world situations.

On the other hand, we find a number of CBM policies based on soft clustering techniques which are able to deal with imperfection. Using fuzzy set theory [13] for uncertainty management, SCBM method [14] denoting *Soft CBM Competence Based Model* was able to handle vagueness in real data by applying foremost the soft clustering technique called Soft DBSCAN-GM (SDG) [15]. Thus, SCBM detects three types of cases in order to maintain case bases by removing noisy and redundant cases. Furthermore, we cite one more CBM policy that tries to deal with all levels of uncertainty in cases, from the complete ignorance to the total certainty, using belief function theory [8] [9]. This approach called ECTD encoding *"Evidential Clustering and case Types Detection for case base maintenance"*. In a nutshell, ECTD approach goes through three main steps: First, it uses the Evidential C-Means (ECM) [16] for the uncertainty management regarding the membership of cases to the different clusters. The partitions centers offered by ECM as well as the different degrees of belief will serve during the second step at the detection of four types of cases: *Noisy cases* have a high degree of belief to not belonging to any one of clusters, *Similar cases* are considered as redundant experiences and situated on the core of the different clusters, *Isolated cases* are situated on clusters borders, so they can only be solved through themselves, and finally *Internal cases* as the representatives of the different clusters. Ultimately, ECTD accomplishes the maintenance by selecting only cases flagged as *Internal* or *Isolated*.

Obviously, ECTD approach [10] follows a good strategy of maintenance with the ability to manage uncertainty. Besides, it showed practically good results. However, this approach is not able to deal with the dynamic aspect of maintenance where cases are grouped according to their solutions with a predefined and static number of clusters. Hence, it does not take into account the dynamicity of the encountered solutions in the case bases knowing that they contain real experiences. In addition, if the case base contains a high number of distinct solutions categories, ECTD approach suffers from a high complexity when dealing with uncertainty towards all possible subsets of solutions. In what follows, we present, therefore, our proposed approach for this paper dealing with these matters, but we offer before that some background related to the belief function theory as well as the used clustering technique.

3 Background: Belief function theory

In order to manage uncertainty in cases as well as the dynamic aspect of maintenance, our contribution is based on Belief function theory and the evidential clustering technique called EK-NNclus. Hence, we show during this section the basic concepts of this theory as well as the corresponding clustering technique.

3.1 Basic concepts

Belief function theory [8][9] is one of the most used theoretical frameworks for reasoning under uncertainty. It is based on the explicit representation and combination of pieces of evidence. Thus, the problem domain is represented through the frame of discernment (Θ) and containing a finite set of elementary events. Hence, each variable ω takes values from Θ. In this theory, a *mass function* m represents the uncertain evidence about ω on Θ. Actually, m is the mapping function from the powerset of Θ containing all possible subsets, denoted 2^{Θ}, to $[0, 1]$ such that:

$$\sum_{A \subseteq \Theta} m(A) = 1 \tag{1}$$

where each mass $m(A)$ is the evidence that supports exactly the ascertain $\omega \in A$. In particular, if $A = \Theta$, $m(\Theta)$ is interpreted as the probability that the evidence does not give us any information about the variable ω from the frame of discernment. If $m(A) > 0$, the event A is called a focal element.

Given a mass function m, the corresponding belief (bel) and plausibility (pl) functions from 2^{Θ} to $[0, 1]$ are defined such that:

$$bel(A) = \sum_{\emptyset \neq B \subseteq A} m(B) \qquad \forall A \subseteq \Theta \tag{2}$$

and

$$pl(A) = \sum_{A \cap B \neq \emptyset} m(B) \qquad \forall A \subseteq \Theta \tag{3}$$

Actually, $bel(A)$ represents the entire belief allocated to support only the event A. However, $pl(A)$ measures the maximum amount of belief that can be assigned to A.

Within belief function theory, several combination rules of evidences can be used. Dempster's rule of combination [9] is one of the most used ones to combine two pieces of evidence (m_1 and m_2) induced from two independent and reliable sources of information. This rule is defined as follows:

$$(m_1 \oplus m_2)(A) = \frac{1}{1-\kappa} \sum_{B \cap C = A} m_1(B)\ m_2(C) \tag{4}$$

where κ is called the conflict of the global combination and defined such that:

$$\kappa = \sum_{A \cap B = \emptyset} m_1(A)\ m_2(B) \tag{5}$$

One of the techniques that allow us to make decision within the belief function framework is the pignistic probabilities transformation and defined as follows:

$$BetP(A) = \sum_{B \subseteq \Theta} \frac{|A \cap B|}{|B|} \frac{m(B)}{1 - m(\emptyset)} \qquad \forall A \in \Theta \tag{6}$$

The decision making is therefore done through the variable having the highest pignistic probability.

3.2 Evidential clustering: E*K*-NNclus

Evidential clustering techniques are aiming to generate credal partition for managing uncertainty in cases' membership to clusters. Among the most known ones, we cite Evidential c-means [16], EVCLUS [18] and EK-NNclus [7] which is based on EKNN rule [19].

The Evidential K-Nearest Neighbor (E*K*NN) rule: In EKNN rule, the knowledge that an object o is distant from an object o_j with a value d_j produces the following piece of evidence m_j on $\Theta = \{\omega_1, ..., \omega_c\}$:

$$m_j(\{\omega_k\}) = u_{jk}\varphi(d_j)\ , \qquad k = 1, ..., c \tag{7a}$$

$$m_j(\Theta) = 1 - \varphi(d_j) \tag{7b}$$

where $\lim_{d \to \infty} \varphi(d) = 0$, and $u_{jk} = 1$ if o_j is classified in ω_k and 0 otherwise.
The K mass functions obtained through the K nearest neighbors are combined then using Dempster's rule as defined in Equations 4 and 5. To make decision about the membership of cases, the combined contour function for $l = 1, .., c$ is defined such that:

$$pl(\omega_l) \propto \prod_{j \in N_K} (1 - \varphi(d_j))^{1-u_{jl}} \tag{8}$$

and its logarithm can be written as follows:

$$ln\ pl(\omega_l) = \sum_{j=1}^{n} v_j\ u_{jl} + C \tag{9}$$

where N_K denotes the indices set of K nearest neighbors, C is a constant, $v_j = -ln(1 - \varphi(d_j))$ if $j \in N_k$, and $v_j = 0$ otherwise.

EK-NNclus: The EK-NNclus algorithm is a decision-directed clustering procedure based on the above EKNN rule. For the initialization step, EK-NNclus starts with a randomly labeled objects (each object exists lonely in one cluster if the number of objects n is not too large, otherwise the number of clusters c will be taken large but lower than n). For objects membership to clusters, we initialize u_{ik} to 1 if the object o_i belongs to cluster k and to 0 otherwise.
Using EKNN rule, the algorithm's iteration updates the object labels in some randomly order. Then, using Equation 9, the logarithms of the membership plausibilities of each object to each cluster are computed as:

$$u_{ik} = \sum_{j \in N_K(i)} v_{ij} s_{jk}, \qquad k = 1, .., c \tag{10}$$

where $N_K(i)$ is the set of indices of the K-NNs of the object o_i. The membership of the object o_i is then updated according to the highest plausibility such that:

$$s_{ik} = \begin{cases} 1 & if\ u_{ik} = \max_{k'} u_{ik'} \\ 0 & Otherwise. \end{cases} \tag{11}$$

A new iteration is started with a randomly reordered objects if exists at least one object that its label changes. In addition, we note a disappearance of clusters from an iteration to another. Finally, and after the convergence of the number of clusters (demonstrated in [7]), the resulting mass function is computed as:

$$m_i = \bigoplus_{j \in N_K(i)} m_{ij} \tag{12}$$

where the different bbas m_{ij} are calculated using Equation 7.

To conclude, EK-NNclus algorithm provides a simpler credal partition compared to other techniques such as ECM [16] and EVCLUS [18] which yield mass functions with 2^{Θ} focal sets. Hence, EK-NNclus is able to avoid the exponential complexity when treating a large number of clusters and it has lower storage requirement. Actually, Ek-NNclus and ECM are equally effective. However, EK-NNclus has an additional major strength which is its automatically determination of clusters number which converges automatically after a number of iterations.

4 DETD: The new proposed CBM method

The purpose of our proposed approach is to use a dynamic clustering technique, which fixes automatically the number of clusters for each case base, in order to select carefully cases that should be maintained. So, what types of cases should be retained in a case base? Actually, we define specially the two following types of cases as the same spirit as those defined in [20]:

- **Isolated cases:** They are far to clusters centers but not noises. Hence, they can only solve themselves and no other cases can solve them. For this reason, their deletion from the case base leads to the decrease of their competence in problem resolution.
- **Internal cases:** Each internal case is a representative of a set of similar cases founded in the same cluster. Hence, deleting all similar cases do not affect the case base competence because they are close to each other. However, internal cases must be maintained to cover all of them.

On the opposite side, there are two other types of cases that should be removed, which are noisy and similar cases. In fact, noisy cases are irrelevant and distort the problem resolution process. Otherwise, similar cases are redundant and useless.

Our newly DETD approach aims to well distinguish between these types of cases while managing uncertainty and dynamicity of new occurred solutions over the time. To do that, we detail in what follows its different steps.

4.1 Step 1: DYNAMIC Evidential Clustering of cases

To partition the case base, this first step of our proposed CBM policy aims at using the clustering algorithm EK-NNclus (see Subsection 3.2) that response to a number of requirements and is characterized with the following properties:

- *Property 1:* Managing the uncertainty in cases' descriptions towards their membership to the different clusters as well as the total ignorance about them.
- *Property 2:* Managing the dynamicity in case bases by fixing automatically the number of clusters through stored cases learning. This property is important to well detecting the different groups of similar cases each time we apply the CBM policy.
- *Property 3:* Managing the scalability while handling uncertainty with a large number of clusters or distinct solutions in case bases.

In our context, EK-NNclus is performed on case bases as our first step in order to generate automatically and dynamically from them the different clusters reflecting cases solutions. Besides, it generates the degrees of belief towards the membership of cases to the different clusters as well as to the partition reflecting the total ignorance (the frame of discernment Θ). Indeed, the output offered by EK-NNclus will be exploited thereafter within case types detection steps.

4.2 Step 2: Isolated cases detection

To distinguish isolated cases from the whole case base, we have foremost to detect noisy cases in order to be eliminated since they seriously affect computations.

Noisy cases detection Actually, uncertainty management and the credal partition offered by EK-NNclus allow us to detect noisy cases, especially through the degrees of belief assigned to the frame of discernment reflecting the complete ignorance. Accordingly, cases having high degree of belief to be assigned to the total ignorance are flagged as noises [7]. Thus, our idea is summed up by detecting noisy cases using the following way:

$$\boldsymbol{o_i} \in NoC \quad iff \quad m_i(\Theta) > \sum_{A_j \subset \Theta} m_i(A_j) \tag{13}$$

where $\boldsymbol{o_i}$ is a case instance and NoC represents the set of all the noisy cases.

Isolated cases detection As it was defined earlier, Isolated cases are situated on the borders of the generated clusters. Hence, we detect them as cases having a distance to the different clusters centers higher than a predefined threshold (do not considering cases that already flagged as noisy -NoC-), otherwise they are considered as similar since they are so close to each others. By this way, we follow these different points:

- The center of each cluster is calculated as the mean of cases attributes values in which they belong. The decision about the membership of cases to the different clusters is achieved using the pignistic probability (Equation 6).
- The threshold for each cluster is fixed as the mean of distances toward its center with excluding cases flagged as noisy.
- To manage uncertainty, also in distances calculation, we compute cases-clusters distances using the Belief Mahalanobis Distance (BMD) as has been used in [10].

Consequently, isolated cases are defined such that:

$$\boldsymbol{o_i} \in IsC \quad if \quad \forall\, k,\ BMD(\boldsymbol{o_i}, \boldsymbol{v_k}) > Threshold_k \tag{14}$$

where IsC represents the set of isolated cases, $\boldsymbol{o_i}$ is a case instance with $\boldsymbol{o_i} \notin NoC$, and $\boldsymbol{v_k}$ presents the center of cluster k.

4.3 Step 3: Detecting Internal case for each generated cluster

Since an internal case presents a prototype of one cluster, we fix it as the nearest case to the center of each generated cluster. Hence, a case is flagged as internal if it has the shortest Belief Mahalanobis Distance (BMD) [10] to one cluster's center. Accordingly, we define formally internal cases as follows:

$$\boldsymbol{o_i} \in InC \quad iff \quad \exists\, k;\ \nexists\, \boldsymbol{o_j}\ /\ BMD(\boldsymbol{o_j}, \boldsymbol{v_k}) < BMD(\boldsymbol{o_i}, \boldsymbol{v_k}) \tag{15}$$

where $\boldsymbol{o_i}$ and $\boldsymbol{o_j}$ are two cases instances, $\boldsymbol{v_k}$ is the center of cluster k, BMD presents the Belief Mahalanobis Distance [10] between cases and clusters, and InC presents the set of all internal cases.

4.4 Step 4: Updating the case base

By arriving to this last step, we have already detected the types of cases that should be selected and maintained for preserving case bases competence in future problem resolution. Therefore, our proposed approach updates ultimately the case base by holding back isolated and internal cases, and removing all the others. By this way, DETD method can be efficient in case bases alleviation while preserving or rather improving their competence in problem resolution.

5 Experimental analysis

In this section, our aim is to evaluate the maintenance quality provided by our approach. Hence, we test it using a number of case bases from U.C.I repository of Machine Learning datasets. While developing, default parameters of EK-NNclus [7] technique are taken. Thus, we propose to measure our maintaining method's effectiveness through three evaluation criteria as done in [10], [11], [12] and [14]. Then, we compare results with those provided by the Initial non-maintained case bases (ICBR) as well as the non-dynamic CBM approach called ECTD [10].

5.1 Evaluation criteria

- **Storage size [S (%)]**: The percentage of the remaining case base's size after maintenance. Hence, it is the rate of case base size reduction, and defined as follows:
$$S = \frac{Number\ of\ cases\ after\ maintenance}{Number\ of\ cases\ before\ maintenance} \times 100$$
- **Time [t (s)]**: The time of problem resolution exerted on 1-Nearest-Neighbor algorithm. This criterion allows to measure the performance of CBR systems in term of retrieval time reduction.
- **Accuracy [PCC (%)]**: The average percentage of correct classification criterion. It is applied using ten fold cross validation runned in front of 1-Nearest-Neighbor as a classification algorithm. Thus, it is defined such that:
$$PCC = \frac{Well\ solved\ problems}{Total\ solved\ problems} \times 100$$

5.2 Dynamic aspect

To evaluate the ability of our approach in handling the dynamic aspect of maintenance, we measure the accuracies dynamically. Drawing to the actual logic provided by CBR systems towards their case bases, we present dynamicity, in our work, through evolving case bases within three consecutive times such that:

- $\boldsymbol{t_1}$: We select randomly from the original case base a subset of cases (CB_1) with respecting the constraint of containing only two solutions (the minimal number that a case base can contain).
- $\boldsymbol{t_2}$: We increment CB_1' size randomly with a probability to meet new solutions (without reaching the entire case base).
- $\boldsymbol{t_3}$: We test on the totality of case bases with the totality of their solutions.

5.3 Results and discussion

According to the evaluation criteria and the dynamicity aspect defined above, we expose the different results in Tables 1, 2 and 3. Actually, in term of reduction size rate (Table 1), our DETD approach has been able to shrink more than half of all the initial tested case bases that contain the totality of cases (100%). For instance, it provides a reduction rate about 35% for "Iris" data set and 37% for "Ionosphere" data set. Obviously, this is the result of redundant and non relevant cases removing. On the other hand, DETD and ECTD approaches are offering very close reduction rates which vary from about 35% to 50% for both of them.

Table 1: Storage size (S%)

Case bases		Storage size (S%)		
		ICBR	ECTD	DETD
1	Glass	100 %	50 %	46.1 %
2	Indian	100 %	51.21 %	43.22 %
3	Ionosphere	100 %	41.03 %	37.42 %
4	Iris	100 %	38.67 %	35.33 %
5	Vehicle	100 %	46.12 %	48.93 %
6	Heberman	100 %	39.87 %	47.07 %

For the retrieval time criterion, it is basically in relation with case bases density involving the storage size criterion. Actually, as shown in Table 2, the retrieval time is remarkably decreasing with the different case bases. For example, it moves on from 0.2825s with ICBR to 0.0062 with the DETD for the "Heberman" data set. Even comparing with ECTD approach, we are noting a slight decreasing of time provided for almost all the different tested case bases.

Table 2: Retrieval time (s)

Case bases		Retrieval time (s)		
		ICBR	ECTD	DETD
1	Glass	0.0091	0.0050	0.0045
2	Indian	0.0125	0.0101	0.0083
3	Ionosphere	0.0156	0.0077	0.0057
4	Iris	0.0841	0.0068	0.0041
5	Vehicle	0.0716	0.0063	0.0061
6	Heberman	0.2825	0.0133	0.0062

Once our newly approach is able to reduce the case base along with the retrieval time and improves accordingly the CBR systems performance, we should now ascertain to their competence stability in problem resolution through the most important criterion called "Accuracy (PCC)". Actually, its results are shown in Table 3 within a dynamic way as defined in Subsection 5.2. For ECTD approach, we fix the number of clusters equal to the number of solutions appearing during t_1 ($K = 2$). Indeed, we note that our DETD approach is more able to preserve the competence of CBs, each time we maintain them. For instance, the accuracy provided by DETD while maintaining "Vehicle" data set is almost stable as it moves on from 74.43% (t_1) to 74.05% (t_2) until 73.55% (t_3). However, it is more and more decreasing after applying ECTD (from about 75% (t_1) until 65% (t_3)). In fact, this is logically explained by the capacity of DETD to handle automatically the number of clusters whereas it is fixed for ECTD and not able to take into account this evolution. Furthermore, we note that DETD and ECTD are providing almost the same results for "Indian", "Ionosphere" and "Heberman" datasets. In fact, it is quite reasonable since they are binary, so the dynamicity in solutions is not really introduced for them. On the other hand, with considering the totality of case bases, we note that the maintained case bases with DETD offer precision even better than non-maintained ones (ICBR) such as for "Glass" data set where it reaches more than 25% of difference in t_2.

Table 3: Dynamic aspect influence in maintenance efficiency

	Case Bases (CB)	Dynamic Accuracy Evaluation (PCC %)								
		t_1			t_2			t_3		
		ICBR	ECTD	DETD	ICBR	ECTD	DETD	ICBR	ECTD	DETD
1	Glass	76.11	77.15	80.1	61.14	70.4	88.24	86.92	63.64	**94.39**
2	Indian	75.98	76.33	76.64	73.57	73.22	74.04	**75.91**	73.78	73.8
3	Ionosphere	78.45	91.03	88.04	82.76	69.71	86.38	86.89	**87.5**	**87.5**
4	Iris	100	99.18	98.54	98.75	96.3	99.02	98	92.21	**98.28**
5	Vehicle	75.12	74.56	74.43	71.89	68.15	74.05	72.34	65.21	**73.55**
6	Heberman	68.76	77.16	75.36	69.2	59.68	73.45	74.18	**76.23**	**76.23**

6 Conclusion

In this paper, we proposed a new case base maintenance method called DETD approach where we focused on exceeding a limitation of some existing CBM policies related to the dynamicity in maintenance. While achieving its main purpose of case base maintenance, our DETD method was able to manage both of uncertainty in cases descriptions using belief function theory tools, and the aspect of dynamicity in CBR systems' case bases using a clustering method that offers a dynamic number of clusters each time we perform the maintenance. Hence, the main idea of our work is summed up by selecting as well as possible only the most relevant types of cases for preserving CBR systems competence and performance. This is actually done, also, through handling uncertainty regarding cases positions towards the different generated clusters. As future work, we aim to propose a dynamic CBM approach in term of real-time mode of maintenance.

References

1. Aamodt, A., Plaza, E.: Case-based reasoning: Foundational issues, methodological variations, and system approaches. *Artificial Intelligence Communications*, vol. 7, 39–52 (1994)
2. Malek, M., Rialle, V.: Design of a case-based reasoning system applied to neuropathy diagnosis. *In European Workshop on Advances in Case-Based Reasoning*, 255–265 (1994).
3. Maher, M. Garza, A.: Case-based reasoning in design. *International journal of IEEE Intelligent Systems*, 34–41 (1997).
4. Kang, B. Yoshida, K. Motoda, H. Compton, P.: Help desk system with intelligent interface. *International journal of Applied Artificial Intelligence*, 611–631 (1997)
5. Varshavskii, P. Eremeev, A.: Analogy-based search for solutions in intelligent systems of decision support integrated models and flexible calculations in artificial intelligence. *International Journal of Computer and Systems Sciences*, 90–101 (2005)
6. Wilson, D. C., Leake, D. B.: Maintaining case-based reasoners: Dimensions and directions. *In Computational Intelligence*, 196–213 (2001)
7. Denoeux, T., Kanjanatarakul, O., Sriboonchitta, S.: EK-NNclus: A clustering procedure based on the evidential K-nearest neighbor rule. *Knowledge-Based Systems.* 88, 57–69 (2015)
8. Dempster, A. P., Upper and lower probabilities induced by a multivalued mapping. *The annals of mathematical statistics*, 325–339 (1967)
9. Shafer, G.: A mathematical theory of evidence. *Vol. 1. Princeton: Princeton university press* (1976)
10. Ben Ayed, S., Elouedi, Z., Lefevre,. E.: ECTD: Evidential clustering and case Types Detection for case base maintenance. *In the 14th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA)*, 1462–1469 (2017)
11. Smiti, A., Elouedi, Z.: COID: Maintaining case method based on Clustering, Outliers and Internal Detection. *In Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, 39–52 (2010)
12. Smiti, A., Elouedi, Z.: WCOID: Maintaining Case-Based Reasoning systems using Weighting, Clustering, Outliers and Internal cases Detection. *In the 11th International Conference on Intelligent Systems Design and Applications (ISDA)*, 356–361 (2011)
13. Zadeh, L.A.: Fuzzy sets. Information and control 8, 338–353 (1965)
14. Smiti, A., Elouedi, Z.: SCBM: Soft Case Base Maintenance method based on competence model. *Journal of Computational Science*, (2017)
15. Smiti, A., Elouedi, Z.: Fuzzy density based clustering method: Soft DBSCAN-GM. *In the 8th International Conference on Intelligent Systems (IS)*, 443–448 (2016)
16. Masson, M. H., Denœux, T.: ECM: An evidential version of the fuzzy c-means algorithm. *Pattern Recognition 41*, 1384–1397 (2008)
17. Smets, P.: The transferable belief model for quantified belief representation. *In Quantified Representation of Uncertainty and Imprecision*, 267–301 (1998)
18. Denœux, T., Masson, M. H.: EVCLUS: Evidential clustering of proximity data. *IEEE Trans. on Systems, Man and Cybernetics B*, 95–109 (2004)
19. Denœux, T.: A k-nearest neighbor classification rule based on Dempster-Shafer theory. *IEEE Trans. on Systems, Man and Cybernetics*, 804–813 (1995)
20. Smiti, A., Elouedi, Z.: WCOID-DG: An approach for case base maintenance based on Weighting, Clustering, Outliers, Internal Detection and DBSCAN-Gmeans. *Journal of computer and system sciences, 80(1)*, 27–38 (2014)