Belief Functions and Cluster Ensembles

Marie-Hélène Masson[1] and Thierry Denoeux[2]

[1] Université de Picardie Jules Verne, IUT de l'Oise
[2] Université de Technologie de Compiègne,
Laboratoire Heudiasyc, BP 20529,
60205 Compiègne, France

Abstract. In this paper, belief functions, defined on the lattice of partitions of a set of objects, are investigated as a suitable framework for combining multiple clusterings. We first show how to represent clustering results as masses of evidence allocated to partitions. Then a consensus belief function is obtained using a suitable combination rule. Tools for synthesizing the results are also proposed. The approach is illustrated using two data sets.

1 Introduction

Ensemble clustering methods aim at combining multiple clustering solutions into a single one, the *consensus*, to produce a more accurate clustering of the data. Several studies have been published on this subject for many years (see, for example, the special issue of the Journal of Classification devoted to the "Comparison and Consensus of Classifications" published in 1986 [4]). The recent interest of the machine learning and artificial intelligence communities for ensemble techniques in clustering can be explained by the success of such ensemble techniques in a supervised context. As recalled in [12,13], various ways of generating cluster ensembles have been proposed. We may use different clustering algorithms or the same algorithm while varying a characteristic of the method (starting values, number of clusters, hyperparameter) [9]. We may also resample the data set [8]. This approach is called *bagged* clustering. Another well-known application of cluster ensembles is called distributed clustering, which refers to the fact that clusterings are performed using different (overlapping or disjoint) subsets of features [21,22,1]. A member of the ensemble is called a *clusterer*. Once several partitions are available, they have to be aggregated into a single one, providing a better description of the data than individual partitions. A variety of strategies have been proposed to achieve this goal: voting schemes [7], hypergraph partitioning [21], pairwise or co-occurrence approach [9,11]. This last approach, which will be shown to have some connections with what is proposed in this paper, is perhaps the simplest approach. The collection of partitions can be be mapped into a squared co-association matrix where each cell (i, j) represents the fraction of times the pair of objects (x_i, x_j) has been assigned to the same cluster. This matrix is then considered as a similarity matrix which can be in turn clustered. A hierarchical clustering algorithm is the most common algorithm used for this purpose.

C. Sossai and G. Chemello (Eds.): ECSQARU 2009, LNAI 5590, pp. 323–334, 2009.

In this paper, we propose a new approach based on belief functions theory. This theory has been already successfully applied to unsupervised learning problems [15,6,16,17]. In those methods, belief functions are defined on the set of possible clusters, the focal elements being subsets of this frame of discernment. The idea here is radically different. It consists in defining and manipulating belief functions on the set of all partitions of the data set. Each clustering algorithm is considered as a source providing an opinion about the unknown partition of the objects. The information of the different sources are converted into masses of evidence allocated to partitions. These masses can be combined and synthesized using some generalizations of classical tools of the belief functions theory.

The rest of the paper is organized as follows. Section 2 gives necessary backgrounds about partitions of a finite set and belief functions defined on the lattice of partitions of a finite set. Section 3 describes how to generate the belief functions, how to combine them and how to synthesize the results. The methodology is illustrated using a simple example. The results of some experiments are shown in Section 4. Finally, Section 5 concludes this paper.

2 Background

2.1 Partitions of a Finite Set

Let E denote a finite set of n objects. A partition p is a set of non empty subsets E_1,...,E_k of E such that:

1) the union of all elements of p, called clusters, is equal to E;
2) the elements of p are pairwise disjoint.

Every partition can be associated to an equivalence relation (i.e., a reflexive, symmetric, and transitive binary relation), on E, denoted by R_p, and characterized, $\forall x, y \in E$, by:

$$R_p(x,y) = \begin{cases} 1 & \text{if } x \text{ and } y \text{ belong to the same cluster in } p \\ 0 & \text{otherwise.} \end{cases}$$

Example. Let $E = \{1,2,3,4,5\}$. A partition p of E, composed of two clusters, the clusters of which are $\{1,2,3\}$ and $\{4,5\}$ will be denoted as $p = (123/45)$. The associated equivalence relation is:

$$R_p = \begin{pmatrix} 1&1&1&0&0 \\ 1&1&1&0&0 \\ 1&1&1&0&0 \\ 0&0&0&1&1 \\ 0&0&0&1&1 \end{pmatrix}.$$

The set of all partitions of E, denoted by $\mathcal{P}(E)$, can be partially ordered using the following ordering relation: a partition p is said to be *finer* than a partition p' on the same set E (or, equivalently p' is *coarser* than p) if the clusters of p

can be obtained by splitting those of p' (or equivalently, if each cluster of p' is the union of some clusters of p). In, this case, we write:

$$p \preceq p'.$$

Note that this ordering can be alternatively defined using the equivalence relations associated to p and p':

$$p \preceq p' \Leftrightarrow R_p(x,y) \leq R_{p'}(x,y) \quad \forall (x,y) \in E^2.$$

The *finest* partition in the order $(\mathcal{P}(E), \preceq)$, denoted $p_0 = (1/2/.../n)$, is the partition where each object is a cluster. The *coarsest* partition is $p_E = (123..n)$, where all objects are put in the same cluster. Each partition precedes in this order every partition derived from it by aggregating two of its clusters. Similarly, each partition succeeds (*covers*) all partitions derived by subdividing one of its clusters in two clusters. The *atoms* of $(\mathcal{P}(E), \preceq)$ are the partitions preceded by p_0. There are $n(n-1)/2$ such partitions, each one having $(n-1)$ clusters with one and only one cluster composed of two objects. Atoms are associated to matrices R_p with only one off-diagonal entry equal to 1.

2.2 Lattice of the Partitions of a Finite Set

The set $\mathcal{P}(E)$ endowed with the $\preceq$-order has a lattice structure [18]. *Meet* ($\wedge$) and *join* ($\vee$) operations can be defined as follows. The partition $p \wedge p'$, called the *infimum* of p and p', is defined as the coarsest partition among all partitions finer than p and p'. The clusters of $p \wedge p'$ are obtained by considering pairwise intersections between clusters of p and p'. The equivalence relation $R_{p \wedge p'}$ is simply obtained by taking the minimum of R_p and $R_{p'}$. The partition $p \vee p'$, called the *supremum* of p and p', is similarly defined as the finest partition among the ones that are coarser than p and p'. The equivalence relation $R_{p \vee p'}$ is given by the *transitive closure* of the maximum of R_p and $R_{p'}$.

2.3 Belief Functions on the Lattice of Partitions

Belief functions [19,20] are most of the time defined on the Boolean lattice of subsets of a finite set. However, following the first investigations of Barthélemy [2], Grabisch [10] has shown that it is possible to extend these notions to the case where the underlying structure is no more the Boolean lattice of subsets, but any lattice. In particular, considering the lattice of partitions, some of the classical constructions and definitions of belief functions (mass assignment, mass combination, commonalities,...) remain valid, up to some adaptations. Let $\mathcal{L} = (\mathcal{P}(E), \preceq)$ denote a lattice of partitions endowed with the meet and join operations defined in section 2.2. A basic belief assignment (bba) is defined as a mass function m from $\mathcal{L}$ to [0;1] verifying:

$$\sum_{p \in \mathcal{L}} m(p) = 1. \tag{1}$$

A bba m is said to be normal if $m(p_0) = 0$. In the rest of this paper, only normal mass functions will be considered. Each partition p that receives a mass $m(p) > 0$ is called a *focal element* of m. A bba m is said to be *categorical* is there is a unique focal element p with $m(p) = 1$. A bba m is said to be of *simple support* if there exists $p \in \mathcal{L}$ and $w \in [0;1]$ such that $m(p) = 1 - w$ and $m(p_E) = w$, all other masses being zero. The bba m can be equivalently represented by a credibility function bel, and a commonality function q defined, respectively, by:

$$\mathrm{bel}(p) \triangleq \sum_{p' \preceq p} m(p'), \tag{2}$$

$$\mathrm{q}(p) \triangleq \sum_{p \preceq p'} m(p'), \tag{3}$$

$\forall p \in \mathcal{L}$. When the reliability of a source (e.g., a clustering algorithm) is doubtful, the mass provided by this source can be discounted using the following operation (discounting process):

$$\begin{cases} m^{\alpha}(p) = (1-\alpha)m(p) \quad \forall p \neq p_E \in \mathcal{L}, \\ m^{\alpha}(p_E) = (1-\alpha)m(p_E) + \alpha, \end{cases} \tag{4}$$

where $0 \leq \alpha \leq 1$ is the discount rate. This discount rate is related to the confidence held by an external agent in the reliability of the source.

Two bbas m_1 and m_2 induced by distinct items of evidence on $\mathcal{L}$ can be combined using the normalized Dempster's rule of combination. The resulting mass function $m_1 \oplus m_2$ will be defined by:

$$(m_1 \oplus m_2)(p) \triangleq \frac{1}{1-K} \sum_{p' \wedge p'' = p} m_1(p')m_2(p'') \quad \forall p \in \mathcal{L}, p \neq p_0 \tag{5}$$

with

$$K = \sum_{p' \wedge p'' = p_0} m_1(p')m_2(p''). \tag{6}$$

Alternatively, one may use the average of m_1 and m_2 defined by:

$$(m_{av})(p) \triangleq \frac{1}{2}\left(m_1(p) + m_2(p)\right) \quad \forall p \in \mathcal{L}. \tag{7}$$

3 Ensemble Clustering

3.1 General Approach

Belief functions, as defined in the previous section, offer a general framework for combining and synthesizing the results of several clustering algorithms. We propose to use the following strategy for ensemble clustering:

1) Mass generation: Given r clusterers, build a collection of r bbas m_1,m_2,...,m_r;
2) Aggregation: Combine the r bbas into a single one using an appropriate combination rule;
3) Synthesis: Provide a summary of the results.

Mass generation. This step depends on the clustering algorithm used to build the ensemble. The simplest situation is encountered when a clusterer produces a single partition p of the data set. To account for the uncertainty of the clustering process, this categorical opinion can be transformed into a simple support mass function using the discounting operation (4). We propose to relate the discounting factor of the source to a cluster validity index, measuring the quality of the partition. Various cluster validity indices can be used for this purpose (see, for instance, [23] for a review of fuzzy cluster validity). In the experiments reported in Section 4, we have used the fuzzy c-means algorithm (converting the fuzzy partition into a hard one) and a partition entropy to define the discounting factor as follows. Let μ_{jk} denote the fuzzy membership degree assigned to the j*th* object and the k*th* cluster and c denote the number of clusters (note that c may vary from a clusterer to another). The normalized partition entropy is a value $0 \leq h \leq 1$ defined by:

$$h = \frac{1}{n \log(c)} \sum_{j=1}^{n} \sum_{k=1}^{c} \mu_{jk} \log(\mu_{jk}). \tag{8}$$

This quantity is maximal (equal to 1) when the quality of the partition is poor, i.e., when all membership values are equal to $1/c$. This value can be used as a discounting factor of the clusterer. This strategy leads to the generation of a bba m, with two focal elements, defined by:

$$\begin{cases} m(p) &= 1 - h, \\ m(p_E) &= h. \end{cases} \tag{9}$$

Suppose now that a clusterer expresses its opinion by a hierarchical clustering [14]. This kind of algorithm produces a sequence of nested partitions, $p_0 \preceq ... \preceq p_E$. At each intermediate stage, the method joins together the two closest clusters. In the *single linkage* approach, the distance between clusters is defined as the distance between the closest pair of objects. The result of this algorithm is commonly displayed as a dendrogram (or classification tree) such as represented in Figure 1. The aggregation levels, used for the representation of the dendrogram, are usually equal to the distances computed when merging two clusters. Note that aggregation levels may be normalized so that the first level is 0 and the last level is 1. This normalization will be assumed in the sequel. Cutting

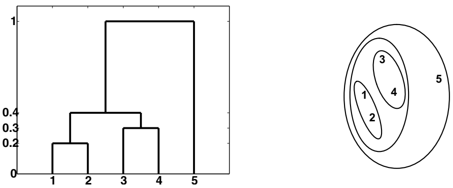


Fig. 1. Example of a small dendrogram (left) and associated nested partitions (right)

the tree at different levels of the hierarchy gives a sequence of nested partitions $\{p_0, p_1, p_2, ..., p_K\}$ (a chain in the lattice of the partitions) associated to levels $\{0, i_1, i_2, ..., i_K\}$ with $p_K = p_E$ and $i_K = 1$. The masses associated to the different partitions can be computed from the levels of the hierarchy as follows. The size of a step between two consecutive levels of the hierarchy in the dendrogram is often considered as an indication on the appropriate number of clusters and on the height at which the dendrogram should be cut. So, we propose the following mass allocation:

$$\begin{cases} m(p_E) = 0 \\ m(p_k) = i_{k+1} - i_k \quad k = 0, \ldots, K-1. \end{cases} \tag{10}$$

For example, cutting the dendrogram of Fig. 1 at different levels produces the bba given in Table 1. It can be seen that the highest mass is allocated to the partition that seems the most natural with respect to the shape of the dendrogram.

Table 1. Bba derived from the dendrogram of Fig. 1.

k	p_k	i_k	$m(p_k)$
0	$p_0 = (1/2/3/4/5)$	0	0.2
1	$p_1 = (12/345)$	0.2	0.1
2	$p_2 = (12/34/5)$	0.3	0.1
3	$p_3 = (1234/5)$	0.4	0.6
4	$p_4 = (12345)$	1	0

Note that many other clustering methods can be described in the same framework. For instance, fuzzy equivalence relations, used for cluster analysis [3], are naturally represented by consonant belief functions on the lattice of partitions.

Combination and synthesis. Once r bbas are available, they can be aggregated into a single one using one of the combination rules recalled in Section 2.3. The interpretation of the results is a more difficult problem, since, depending on the number of clusterers in the ensemble, on their nature anf the conflict between them, and on the combination rule, a potentially high number of focal elements may be found. If the number of focal elements in the combined bba is too high to be explored, a first way to proceed is to select only the partitions associated with the highest masses or use a simplification algorithm such as described in [5]. We propose another approach which consists in building a matrix $Q = (q_{ij})$ whose elements are the commonalities associated to each atom of the lattice of partitions. This approach amounts computing, for each pair of object (i, j), a new similarity measure q_{ij} by accumulating the masses which support the association between i and j:

$$q_{ij} = \sum_p m(p) R_p(i, j). \tag{11}$$

Matrix Q can be in turn clustered using, for instance, a hierarchical clustering algorithm. If a partition is needed, the classification tree can be cut at a specified level or so as to insure a user-defined number of clusters. Note that the co-association method proposed in [9] is recovered as a special case of our approach if the consensus has been obtained by averaging the masses of the individual clusterers.

3.2 Toy Example

Let $E = \{1, 2, 3, 4, 5\}$ be a set composed of 5 objects. We assume that two clustering algorithms have produced partitions $p_1 = (123/45)$ and $p_2 = (12/345)$. As it can be seen, the partitions disagree on the third element which is clustered with $\{1, 2\}$ in p_1 and $\{4, 5\}$ in p_2. As proposed in Section 3.1, we construct two simple mass functions by discounting each clusterer i by a factor α_i. In a first situation, we consider that we have an equal confidence in the two clusterers, so we fix $\alpha_1 = \alpha_2 = 0.1$. We have:

$$m_1(p_1) = m_2(p_2) = 0.9 \quad m_1(p_E) = m_2(p_E) = 0.1,$$

with $p_E = (12345)$. Applying Dempster's rule of combination (5)-(6) leads to the following combined bba $m = m_1 \textcircled{\cap} m_2$:

Focal elements	mass m	bel
$p_1 \wedge p_2 = (12/3/45)$	0.81	0.81
$p_1 = (123/45)$	0.09	0.90
$p_2 = (12/345)$	0.09	0.90
$p_E = (12345)$	0.01	1

Suppose now that the confidence is less in the second clusterer than in the first one. We fix $\alpha_1 = 0.1$ and $\alpha_2 = 0.2$. In that case, we obtain a bba m' characterized by:

Focal elements	mass m'	bel'
$p_1 \wedge p_2 = (12/3/45)$	0.72	0.72
$p_1 = (123/45)$	0.18	0.90
$p_2 = (12/345)$	0.08	0.80
$p_E = (12345)$	0.02	1

The commonalities of the atoms of the lattice are given for the two situations by the following matrices:

$$Q = \begin{pmatrix} 1 & 1 & 0.1 & 0.01 & 0.01 \\ 1 & 1 & 0.1 & 0.01 & 0.01 \\ 0.1 & 0.1 & 1 & 0.1 & 0.1 \\ 0.01 & 0.01 & 0.1 & 1 & 1 \\ 0.01 & 0.01 & 0.1 & 1 & 1 \end{pmatrix} \quad Q' = \begin{pmatrix} 1 & 1 & 0.2 & 0.02 & 0.02 \\ 1 & 1 & 0.2 & 0.02 & 0.02 \\ 0.2 & 0.2 & 1 & 0.1 & 0.1 \\ 0.02 & 0.02 & 0.1 & 1 & 1 \\ 0.02 & 0.02 & 0.1 & 1 & 1 \end{pmatrix}$$

Applying the single linkage algorithm to these two matrices gives the hierarchical clusterings represented in Figure 2. The dendrogram may be seen as a good synthesis of the information (consensual and conflicting) provided by the clusterers. On the left, we can see that no cut is able to recover a partition in which object 3 is associated to the other objects (except in the root of the tree). On the right, cutting the tree at a level greater than 0.8, allows us to recover the partition given by m_1, reflecting the fact that a greater confidence is allocated to this source.

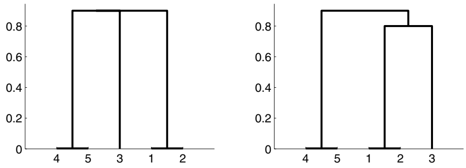


Fig. 2. Dendrograms computed from Q (left) and Q' (right) for the toy example

4 Two Examples

4.1 Distributed Clustering

In a distributed computing environment, the data set is spread into a number of different sites. In that case, each clusterer has access to a limited number of features and the distributed computing entities share only higher level information describing the structure of the data such as cluster labels. The problem is to find a clustering compatible with what could be found if the whole set of features was considered. To illustrate this point, we used a dataset named 8D5K found in [21]. This dataset is composed of five Gaussian clusters in dimension 8. Out of the 1000 points of the original data set, we retain only 200 points (40 points per cluster). We created five 2D views of the data by selecting five pairs of features. We applied the fuzzy c-means algorithm in each view (each one with $c = 5$) to obtain five hard partitions computed from the fuzzy partitions. These partitions are represented in Figure 3. The left row shows the partitions in the 2D views, and the right row shows the same partitions projected onto the first two principal components of the data. An ensemble of five mass functions was constructed using the approach proposed in Section 3.1: each clusterer, discounted according the entropy of partition (8), was represented by a mass function with two focal elements. A "consensus" clustering was obtained by applying Dempster's rule of combination, computing the matrix Q and the associated tree using single linkage, and cutting the tree to obtain five clusters. The consensus clustering is presented in Figure 3. It may be seen that a very good clustering is obtained, although some of the partitions provided by the clusterers were poor in the space described by the eight features.

4.2 Non Elliptical Clusters

This section is intended to show that the proposed approach is able to detect clusters with complex shapes. The half-ring data set is inspired from [9]. It

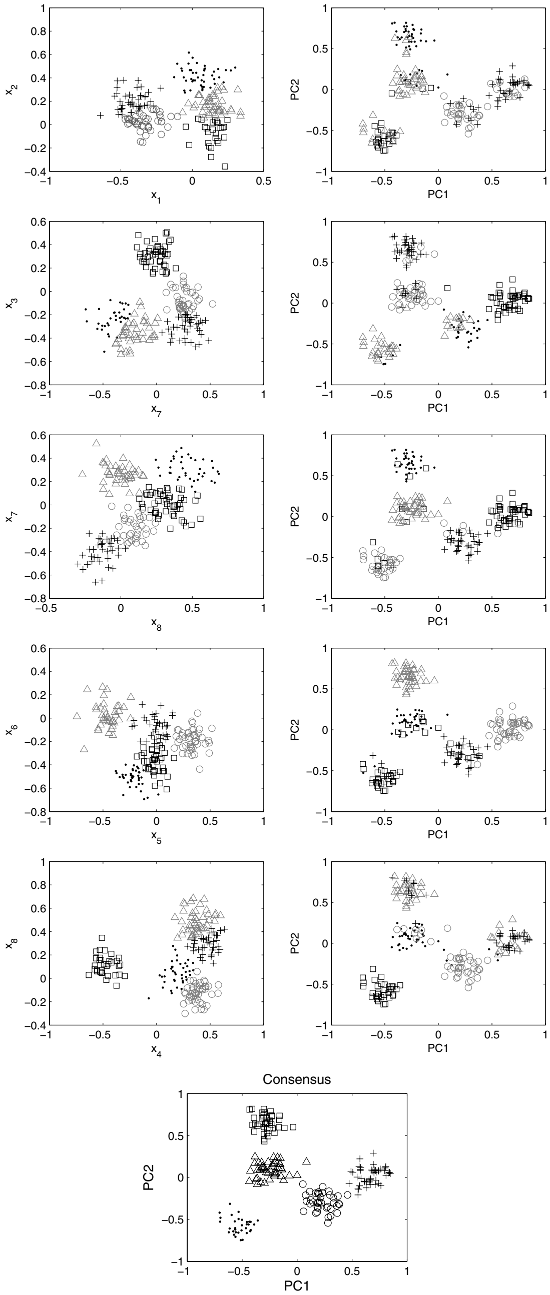
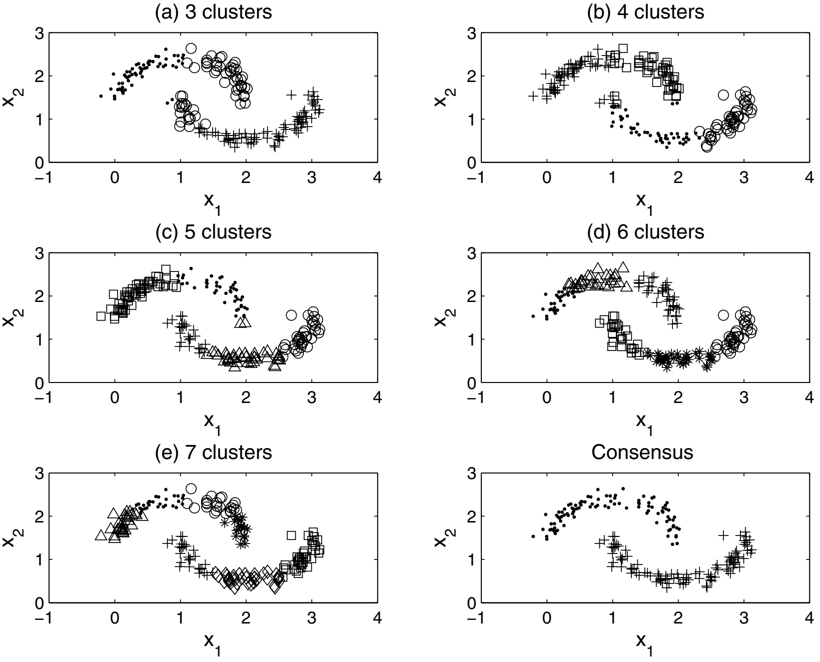


Fig. 3. 8D5K data set [21]. The ensemble is composed of five individual clustering solutions obtained from five 2D views of the data. The left row shows the partition obtained in each two-dimensional features space and the right row shows the corresponding partition in the plane spanned by the two first principal components.

consists of two clusters of 100 points each in a two-dimensional space. To build the ensemble, we use the fuzzy c-means algorithm with a varying number of clusters (3 to 7). The hard partitions are represented in Figure 4.

As in the previous example, each partition was discounted using the entropy of partition and five mass functions with two focal elements each were combined using Dempster's rule of combination. A tree was computed from the commonality matrix using the single linkage algorithm and a partition in two clusters was derived from the tree. This partition is also represented in Figure 4. We can see that the natural structure of the data is perfectly recovered.


(a) 3 clusters

(b) 4 clusters

(c) 5 clusters

(d) 6 clusters

(e) 7 clusters

Consensus

Fig. 4. Half-rings data set. Figures (a) to (e) show the individual clusterings in the 2D views; the last figure shows the consensual clustering.

5 Conclusion

We have proposed in this paper a new approach for aggregating multiple clusterings. This approach is based on the use of belief functions defined on the lattice of partitions of the set of objects to be clustered. In this framework, it possible to assign masses of evidence to partitions. We have shown that a wide variety of clusterers can be naturally represented in this framework and that combination tools can provide a "consensual " description of the data. The preliminary experiments on several data sets have shown the usefulness of the method. A drawback of the conjonctive combination, especially when the conflict between the clusterers is important, is to potentially generate a large number of focal elements. A similar problem was already encountered in [15]. Future work will investigate how to simplify the result by merging similar or unimportant focal elements using a procedure similar to the one proposed in [5].

References

1. Avogadri, R., Valentini, G.: Fuzzy ensemble clustering for DNA microarray data analysis. In: Masulli, F., Mitra, S., Pasi, G. (eds.) WILF 2007. LNCS, vol. 4578, pp. 537–543. Springer, Heidelberg (2007)
2. Barthélemy, J.-P.: Monotone functions on finite lattices: an ordinal approach to capacities, belief and necessity functions. In: Fodor, J., de Baets, B., Perny, P. (eds.) Preferences and decisions under incomplete knowledge, pp. 195–208. Physica-Verlag, Heidelberg (2000)
3. Bezdek, J.C., Keller, J., Krisnapuram, R., Pal, N.R.: Fuzzy Models and Algorithms for Pattern Recognition and Image Processing. Series: The Handbooks of Fuzzy Sets, vol. 4. Kluwer Academic Publishers, Nonwell (1999)
4. Day, W.: Foreword: Comparison and Consensus of Classifications. Journal of Classification 3, 183–185 (1986)
5. Denœux, T.: Inner and outer approximation of belief structures using a hierarchical clustering approach. Int. Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 9(4), 437–460 (2001)
6. Denœux, T., Masson, M.-H.: EVCLUS: EVidential CLUStering of proximity data. IEEE Transactions on Systems, Man and Cybernetics Part B 34(1), 95–109 (2004)
7. Dimitriadou, E., Weingessel, A., Hornik, K.: Voting-Merging: an ensemble method for clustering. In: Dorffner, G., Bischof, H., Hornik, K. (eds.) ICANN 2001. LNCS, vol. 2130, p. 217. Springer, Heidelberg (2001)
8. Dudoit, S., Fridlyand, J.: Bagging to improve the accuracy of a clustering procedure. Bioinformatics 19(9), 1090–1099 (2003)
9. Fred, A., Jain, A.K.: Data clustering using evidence accumulation. In: Proc. of the 16th International Conference on Pattern Recognition, Quebec, Canada, pp. 276–280 (2002)
10. Grabisch, M.: Belief functions on lattices. International Journal of Intelligent Systems 24(1), 76–95 (2009)
11. Greene, D., Tsymbal, A., Bolshakova, N., Cunningham, P.: Ensemble clustering in medical diagnostics. In: Proc. of the 17th IEEE Symposium on Computer-Based Medical Systems, Bethesda, MD, USA, pp. 576–581 (2004)
12. Hadjitodorov, S.T., Kuncheva, L., Todorova, L.: Moderate diversity for better cluster ensemble. Information Fusion 7(3), 264–275 (2006)
13. Hornik, K., Leisch, F.: Ensemble methods for cluster analysis. In: Taudes, A. (ed.) Adaptive Information Systems and Modelling in Economics and Management Science. Interdisciplinary Studies in Economics and Management, pp. 261–268. Springer, Heidelberg (2005)
14. Jain, A.K., Dubes, R.C.: Algorithms for Clustering Data. Prentice Hall, Englewood Clis (1988)
15. Le Hégarat-Mascle, S., Bloch, I., Vidal-Madjar, D.: Application of Dempster-Shafer Evidence Theory to Unsupervised Classification in Multisource Remote Sensing. IEEE Transactions on Geoscience and Remote Sensing 35(4), 1018–1031 (1997)

16. Masson, M.-H., Denœux, T.: Clustering interval-valued data using belief functions. Pattern Recognition Letters 25(2), 163–171 (2004)
17. Masson, M.-H., Denœux, T.: ECM: An evidential version of the fuzzy *c*-means algorithm. Pattern Recognition 41, 1384–1397 (2008)
18. Montjardet, B.: The presence of lattice theory in discrete problems of mathematical social sciences. Why. Mathematical Social Sciences 46, 103–144 (2003)
19. Shafer, G.: A mathematical theory of evidence. Princeton University Press, Princeton (1976)
20. Smets, P., Kennes, R.: The Transferable Belief Model. Artificial Intelligence 66, 191–243 (1994)
21. Strehl, A., Ghosh, J.: Cluster ensemble - a knowledge reuse framework for combining multiple partitions. Journal of Machine Learning Research 3, 563–618 (2002)
22. Topchy, A., Jain, A.K., Punch, W.: Combining multiple weak clusterings. In: Proc. of the IEEE Int. Conference on Data Mining, Melbourne, Australia, pp. 331–338 (2003)
23. Wang, W., Zhang, Y.: On fuzzy cluster validity indices. Fuzzy Sets and Systems 158(19), 2095–2117 (2007)