Martin Hahmann, Peter B. Volk, Frank Rosenthal, Dirk Habich, Wolfgang Lehner

How to Control Clustering Results? Flexible Clustering Aggregation

Erstveröffentlichung in / First published in:

Advances in Intelligent Data Analysis VIII: 8th International Symposium on Intelligent Data Analysis. Lyon, 31.08. - 02.09.2009. Springer, S. 59-70. ISBN 978-3-642-03915-7.

DOI: http://dx.doi.org/10.1007/978-3-642-03915-7_6

Final edited form was published in "Advances in Intelligent Data Analysis VIII: 8th International Symposium on Intelligent Data Analysis. Lyon 2009", S. 59-70, ISBN 978-3-642-03915-7
http://dx.doi.org/10.1007/978-3-642-03915-7_6

How to Control Clustering Results? Flexible Clustering Aggregation

Martin Hahmann, Peter B. Volk, Frank Rosenthal, Dirk Habich, and Wolfgang Lehner

Dresden University of Technology, Database Technology Group
dbinfo@mail.inf.tu-dresden.de

Abstract. One of the most important and challenging questions in the area of clustering is how to choose the best-fitting algorithm and parameterization to obtain an optimal clustering for the considered data. The clustering aggregation concept tries to bypass this problem by generating a set of separate, heterogeneous partitionings of the same data set, from which an aggregate clustering is derived. As of now, almost every existing aggregation approach combines given crisp clusterings on the basis of pair-wise similarities. In this paper, we regard an input set of soft clusterings and show that it contains additional information that is efficiently useable for the aggregation. Our approach introduces an expansion of mentioned pair-wise similarities, allowing control and adjustment of the aggregation process and its result. Our experiments show that our flexible approach offers adaptive results, improved identification of structures and high useability.

1 Introduction

Data clustering is an important data-mining technique, commonly used in various domains [1,2,3]. This technique can be utilized for an initial exploration of scientific data and often builds the basis for subsequent analysis techniques. Generally, clustering is defined as the problem of partitioning a set of objects into groups, so-called clusters, so that objects in the same group are similar, while objects in different groups are dissimilar [3]. Following this definition, a well-defined measure for similarity between objects is required [1,3].

In this area, the selection of the best-fitting clustering algorithm, including parameterization, from the multitude of options is a non-trivial task, especially for users who have little experience in this area. However, this selection issue is vital for the clustering result quality [3] and therefore, users usually conduct the following steps in an iterative way until a satisfying result is achieved: *algorithm selection*, *parameter selection*, *clustering* and *evaluation*. This iterative approach is tedious work and requires profound clustering knowledge. Therefore, an alternative approach to make clustering more applicable for a wide range of non-clustering experts in several domains is desirable.

On a conceptual level this issue can be tackled by applying the clustering aggregation technique. Fundamentally, clustering aggregation combines an ensemble of several partitionings of a data set, generated using different algorithms

Final edited form was published in "Advances in Intelligent Data Analysis VIII: 8th International Symposium on Intelligent Data Analysis. Lyon 2009", S. 59-70, ISBN 978-3-642-03915-7
http://dx.doi.org/10.1007/978-3-642-03915-7_6

and/or parameters, into a final clustering result and hence avoids the fixation on only one clustering method. As demonstrated in various papers, the quality and robustness of the aggregated clustering result increase in comparison with the input clusterings [4,5,6,7]. The proposed aggregation techniques can be classified into three basic classes: **(i)** pair-wise assignment class [7,8,9], **(ii)** hypergraph-based class [7], and **(iii)** cluster correspondence class [6,10,11].

Most aggregation techniques are members of the first class, relying on pair-wise assignments and on an associated majority decision. Typically, they use a set of crisp clustering results as basic input. A crisp result, e.g., one determined by k-means [2,3] or DBSCAN[1], assigns each object exclusively to the cluster having the highest similarity with the object. Therefore, a pair of objects can be located: (i) in the same or (ii) in different clusters. For the final assignment case of a pair in the aggregate, the assignment occurring most in the input set of clusterings is selected. For details about the second class refer to Strehl et al. [7] and to Boulis et al. [6] and Topchy et al. [12] for examples of class three. In comparison, class one utilizes the most information about the data and is thus presumed to be the optimal approach for aggregation at the moment.

The aggregation concept increases the clustering quality as well as the robustness [4,5,6,7] and frees the user from selecting the optimal algorithm and parameterization. But the aggregation process itself is not controllable in an efficient way. If an user obtains an unsatisfying aggregate, the only adjustment option consists of the modification of input clusterings. To allow control and result flexibility, we present an enhanced aggregation concept. The first enhancement concerns the aggregation input. Instead of using crisp results, we utilize soft clustering results—assigning each object its similarity to all determined clusters—obtained via algorithms like *FCM* [13] or refinement techniques like a-posteriori [4]. This fine-grained information is efficiently useable to expand pair-wise assignments making them more accurate. With this second enhancement, we are able to (i) revise the aggregation itself and (ii) introduce user-friendly control options.

To summarize, we propose our novel flexible aggregation concept in this paper. The contributions—also reflecting the structure of the paper—are as follows: We start with a detailed description of already available clustering aggregation concepts in Section 2 and highlight several drawbacks. In Section 3, we expand the pair-wise assignments for soft clusterings and introduce a novel significance score. Based on these expansions, we propose our flexible aggregation method in Section 4. In Section 5, we conduct an exhaustive evaluation and present future research aspects, before we conclude the paper in Section 6.

2 Preliminaries

For the explanations made in this paper, we assume the following **setting:** let $\mathcal{D}$ be a dataset $\{x_1, \cdots, x_n\}$ consisting of n points—also called objects—and $\mathcal{C}$ be a cluster ensemble $\{C_1, \cdots, C_e\}$, created with different algorithms and parameterizations. Each $C_l \in \mathcal{C} (1 \leq l \leq e)$ has k_l clusters $c_1, \cdots, c_{k_l}$, satisfying $\bigcup_{i=1}^{k_l} c_i = \mathcal{D}$. Based on this, the common **goal** is the construction of an aggregate clustering $\hat{C}$ by combining all members of $\mathcal{C}$.

Final edited form was published in "Advances in Intelligent Data Analysis VIII: 8th International Symposium on Intelligent Data Analysis. Lyon 2009", S. 59-70, ISBN 978-3-642-03915-7
http://dx.doi.org/10.1007/978-3-642-03915-7_6

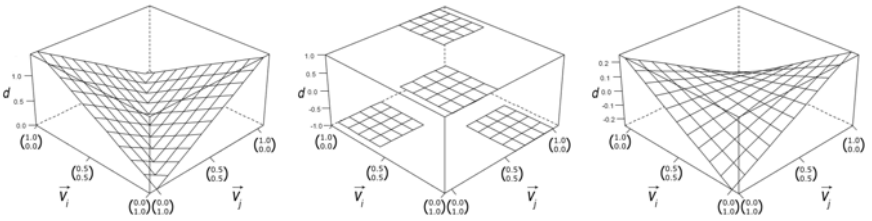
Utilizing a given set of crisp clusterings, each point $x_i \in D$ has a unique label denoting its cluster assignment. Regarding the pair-wise similarities of two points in C_l, two pair-wise assignment cases (*pa*-cases) are definable: (i) $\boldsymbol{a}_+$ for objects with equal cluster labels that are located in the same cluster of C_l and (ii) $\boldsymbol{a}_-$ for object pairs featuring different labels, indicating membership in separate clusters of C_l. To construct $\hat{C}$, the *pa*-case of every pair of points from $\mathcal{D}$ is determined for each clustering of $\mathcal{C}$. After that, the *pa*-case that is dominant throughout $\mathcal{C}$ is selected to hold for the respective pair in $\hat{C}$ [5,7,8,9]. **Example**: two objects $x_1; x_2$ that belong to the same cluster in 7 out of 10 clusterings of $\mathcal{C}$ also belong to the same cluster in $\hat{C}$.

To use soft clusterings as input for the clustering aggregation, we need to update our setting. Each point $x_i \in \mathcal{D}$ is now assigned to all clusters of C_l to a certain degree. Thus, the assignment information of x_i in C_l is denoted as a vector $\vec{v_i}$ with the components $v_{ip} (1 \leq p \leq k_l)$ describing the relation between x_i and the p-th cluster of C_l. Clustering aggregation based on soft assignments is challenging because it requires the determination of *pa*-cases using vectors. We are able to simply adopt the previous approach by stating that x_i and x_j are members of the same cluster if their assignment vectors $\vec{v_i}$ and $\vec{v_j}$ are equal by components. This condition is very strict and would most likely lead to nearly no a_+ assignments. Therefore, this constraint is softened and the a_+ case now holds for objects with similar assignment vectors.

This principle is employed by available aggregation concepts for soft input sets [4,14]. Both approaches use well-known distance measures—e.g. the euclidean distance in [14]—to calculate the similarity between vectors and to derive the *pa*-cases. If the calculated vector similarity exceeds a certain threshold, the resp. points are considered as a_+ or else as a_-. Per definition, the approaches of [4,14] do not deal with aggregation control. Their major problem, described subsequently, concerns the handling of soft assignments, using only common distance measures. For evaluation in this context, we assume the following experimental **setup**: a clustering C_l with $k_l = 2$, a set of 121 vector pairs $\vec{v_i}; \vec{v_j}$, satisfying $\sum_{p/q=1}^{2} v_{ip/jq} = 1, i \neq j, 0 \leq v_{ip/jq} \leq 1$, where $v_{ip/jq}$ are multiples of 0.1.

We start by applying the L_2 norm resp. *euclidean* distance to our setup. In Fig. 1(a), the obtained results are shown; (i) via x- and y-coordinates a vector pairing is specified, while (ii) the corresponding z-value represents the L_2 distance for this pair. **Example**: the pair $\vec{v_i}^\top = (1, 0)$ and $\vec{v_j}^\top (0, 1)$ (western corner of Fig. 1(a)) has a distance of $\sqrt{2}$. Basically, L_2 is a non-directional distance function only considering the norm, which is a major drawback when measuring similarity of vectors in this case. Thus, pairs $\vec{v_i}; \vec{v_j}$ can have equal L_2 distances regardless of x_i and x_j actually being in the same cluster or not. **Example**: the pair $\vec{v_i}^\top = (0.1, 0.9)$ and $\vec{v_j}^\top (0.3, 0.7)$ is located in cluster 2, i.e. a_+ holds; pair $\vec{v_k}^\top = (0.6, 0.4)$; $\vec{v_l}^\top (0.4, 0.6)$ is separated in clusters 1 and 2, i.e. a_-. Although *pa*-cases are actually different, both pairs have the same L_2 distance of $\sqrt{0.08}$. It is obvious that this can lead to incorrect decisions in the construction of $\hat{C}$, especially if thresholds or clustering algorithms are employed. Consequently, vector direction is vital for an accurate interpretation of *pa*-cases.

Final edited form was published in "Advances in Intelligent Data Analysis VIII: 8th International Symposium on Intelligent Data Analysis. Lyon 2009", S. 59-70, ISBN 978-3-642-03915-7
http://dx.doi.org/10.1007/978-3-642-03915-7_6



(a) L2-norm (b) correlation coefficient (c) covariance

Fig. 1. Different distance measures applied to 2-dimensional vectors

Next, we examine distance metrics considering the direction resp. composition of vectors. First, we look at the *Pearson correlation coefficient* (ϱ) assuming a_+ for positive and a_- for negative linear dependency between $\vec{v_i}$ and $\vec{v_j}$. In Fig.1(b), we can see two pairs of separated planes as results of our experiment. When examining vector pairs and their corresponding ϱ, we can confirm our assumption about the relation between the value of $\varrho(\vec{v_i}, \vec{v_j})$ and *pa*-cases. The correlation coefficient has two advantages: (i) direction awareness and (ii) a direct link between the *pa*-case and the algebraic sign of the ϱ-value.

Regarding Fig.1(b), we notice gaps between the planes. These originate from vector pairs where at least one member has zero variance ($\sigma^2 = 0$). The *Pearson correlation coefficient* is defined as the ratio of the covariance of two vectors and the product of their standard deviations. Therefore, $\sigma^2 = 0$ leads to a division by zero, making ϱ undefined. To get rid of this problem, we exclude the mentioned division from ϱ, reducing it to the *covariance*.The results for this last experiment are shown in Fig. 1(c). We observe a behavior similar to ϱ, but in contrast there are no undefined areas and continuous values. The last two experiments have shown a special behavior of ϱ and covariance for vectors with $\sigma^2 = 0$. While ϱ is not defined for these cases, the covariance yields zero.

Vectors $\vec{v_i}$ with σ^2=0 are an interesting phenomenon in our soft clustering scenario. They satisfy $\forall v_{ip} | v_{ip} = \frac{1}{k_l}$, stating that the respective object x_i has equal relations with all clusters of C_l. Thus, it is impossible to determine an explicit cluster affiliation for this object. We refer to such cases as *fully balanced* assignments. Since we cannot decide in which cluster an object x_i with a fully balanced assignment is situated in, it is also impossible to determine a *pa*-case for a pair $x_i; x_j$ if at least one member has a fully balanced assignment. Until now, all clustering aggregation approaches assume only two possible *pa*-cases, a_+ and a_-. With the emergence of fully balanced assignments, a novel additional *pa*-case can be defined covering object pairs with undecidable assignments.

3 Expanding Pair-Wise Assignments

The previous section has shown that the determination of pair-wise assignments is a non-trivial task in a scenario utilizing soft cluster mappings. Existing approaches use common distance functions to solve this problem. But our experiments brought up two major flaws of this concept: **(i)** not all distance functions

Final edited form was published in "Advances in Intelligent Data Analysis VIII: 8th International Symposium on Intelligent Data Analysis. Lyon 2009", S. 59-70, ISBN 978-3-642-03915-7
http://dx.doi.org/10.1007/978-3-642-03915-7_6

can be effectively applied and **(ii)** fully balanced assignments are ignored. In this section we expand the concept of pair-wise assignments to fix the aforementioned problems and introduce a novel significance score for *pa*-cases.

3.1 A Novel Pair-Wise Assignment

In our preliminaries, we described *fully balanced* assignments as a special kind of assignments that make the identification of an explicit cluster relation impossible. Until now, the concept of pair-wise assignments has been restricted to two possible cases that both need definite cluster affiliations. Therefore, proper handling of fully balanced cases requires a novel third assignment case. This case covers undecidable pair-wise assignments and will be denoted as $\boldsymbol{a}_?$. To correctly determine a *pa*-case for any pair $x_i; x_j$ in a clustering C_l, we need to know if a $\vec{v_i}$ is fully balanced. This is the case if each component of $\vec{v_i}$ equals $\frac{1}{k_l}$. An additional form of undecidable assignments, which we denote as *balanced*, occurs with vectors having more than one maximum component v_ip. Assume e.g. an object x_i with $\vec{v_i}^\top = (0.4, 0.4, 0.2)$ for a clustering C_l with $k_l = 3$ clusters. Although we can state that x_i is not a member of cluster 3, it is impossible to specify whether the object effectively belongs to cluster 1 or 2. In contrast, a vector $\vec{v_i}^\top = (0.6, 0.2, 0.2)$ containing multiple equal but not maximal components v_{ip} is not critical. As long as the maximum v_{ip} is singular, we can derive a clear cluster affiliation. Based on this observation, we define a balance-detection function $b(\vec{v_i})$ testing if an object x_i has a fully balanced or a balanced assignment. If $\vec{v_i}$ contains multiple maxima, hence showing no clear cluster affiliation, the function $b(\vec{v_i})$ results in *true*; otherwise $b(\vec{v_i})$ yields *false*.

Next, we need to decide whether x_i and x_j belong to the same partition of C_l or not. Therefore, we regard the strongest cluster affiliation of x_i i.e. the maximum v_{ip}. If the maximum components v_{ip} and v_{jq} of two vectors $\vec{v_i}; \vec{v_j}$, are located in the same dimension of their respective vectors, x_i and x_j belong to the same cluster. In contrast, objects with maximum components in different dimensions of $\vec{v_i}$ are located in different clusters. Based on this, we define a co-occurrence function $c(\vec{v_i}, \vec{v_j})$, stating whether $x_i; x_j$ are part of the same cluster:

$$c(\vec{v_i}, \vec{v_j}) \begin{cases} 1 & \text{if } \{p|v_{ip} = max(\vec{v_i})\} \cap \{q|v_{jq} = max(\vec{v_j})\} \neq \emptyset \\ -1 & \text{otherwise} \end{cases} \tag{1}$$

$$case(x_i, x_j) = \begin{cases} 1, & \text{if } c(\vec{v_i}, \vec{v_j}) = 1 \text{ and } \neg(b(\vec{v_i}) \vee b(\vec{v_j})) \\ -1, & \text{if } c(\vec{v_i}, \vec{v_j}) = -1 \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where $i \neq j$, $1 \leq (p, q) \leq k_l$ and $max(\vec{v_i})$ returns the maximum component of $\vec{v_i}$. Now, we can create a function *case*() (eq.2) that determines the *pa*-case of any object pair in a given clustering C_l. Our function $case(x_i, x_j)$ returns 1 if a_+ holds for x_i and x_j. This is the case if no object has a balanced or fully balanced $\vec{v_i}$ and if both objects are clearly related with the same cluster of C_l. The result -1 denotes the *pa*-case a_-. There, it is not relevant if balanced

Final edited form was published in "Advances in Intelligent Data Analysis VIII: 8th International Symposium on Intelligent Data Analysis. Lyon 2009", S. 59-70, ISBN 978-3-642-03915-7
http://dx.doi.org/10.1007/978-3-642-03915-7_6

objects are part of the pair in question. Assume for C_l with $k_l = 3$ a balanced $\vec{v_i} = (0.4, 0.4, 0.2)$ and $\vec{v_j} = (0.1, 0.1, 0.8)$. Since the maximum components are in different dimensions, a_- holds. Although we cannot decide to which cluster x_i belongs, it is definitely not the cluster x_j belongs to. For undecidable cases like pairs containing fully balanced objects or pairs with balanced assignments that co-occur ($c(\vec{v_i}, \vec{v_j}) = 1$), $case()$ yields 0, indicating $a_?$. Our function $case()$ solves the problems described at the beginning of this section and allows the correct determination of one of our three *pa*-cases for any arbitrary object pair.

3.2 Introducing Significance

By definiton, our novel $a_?$ case is limited to specific vector compositions, whereas the remaining two *pa*-cases apply for nearly all possible object pairs resp. a wide range of $\vec{v_i}$'s. Therefore, it is obvious to bring up the question of significance. In other words, is a decision for a certain pair of objects made with more or less confidence than for other pairs?

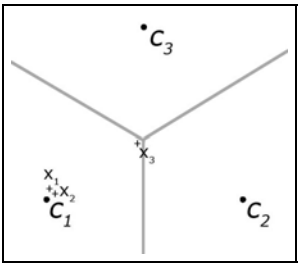


Fig. 2. $\vec{v_i}$ with different significance

Consider the example shown in Fig. 2, of a clustering C_l with $k_l = 3$ clusters and their respective centroids c_1, c_2 and c_3. The grey lines show the borders of the area of influence each cluster has. An object located on those lines or at intersection points has an equal degree of similarity with adjacent clusters and has thus a balanced resp. fully balanced assignment. The two depicted objects x_1 and x_2 have a very strong relation with c_1 and only negligible links with the remaining clusters of C_l. For this example, our function $case(x_1, x_2)$ results in 1, hence a_+ would be stated for x_1 and x_2. Now regard object x_3: it still has the strongest degree of similarity with c_1 but it also has a nearly equal similarity with c_2 and c_3, bringing x_3 very close to a fully balanced assignment. Nevertheless, $case(x_1, x_3)$ determines that x_1 and x_3 both belong to cluster c_1, which is correct in this example. Regarding both resulting *pa*-cases, we would intuitively say that the one made for x_1, x_2 has more confidence.

When the significance of a *pa-case* is evaluated in a subjective way, two properties have to be respected: **(i)** $\vec{v_i}$ and $\vec{v_j}$ should show an explicit cluster relationship i.e. show high dissimilarity to the fully balanced assignment, like $x_1; x_2$ in Fig. 2; **(ii)** $\vec{v_i}$ and $\vec{v_j}$ should have a high component-wise similarity, which is also the case for $x_1; x_2$ in Fig. 2. Examples of pairs maximizing both factors are the corners of the planes, shown in Fig. 1. It is plausible to assume that, starting from these locations, the significance should decrease when approaching the middle of the plane or one of its bisectors (one of the grey lines in Fig. 2 resp.), where balanced or fully balanced assignments are located. As we can see in Fig. 1(c), the covariance partly shows this desired behavior of high values at the corners and low resp. zero values in the middle of the plane. Using this observation, we define a significance measure $s(\vec{v_i}, \vec{v_j})$—similar to the covariance—that returns a significance score for a *pa*-case determined for a pair $x_i; x_j$ in a clustering C_l.

Final edited form was published in "Advances in Intelligent Data Analysis VIII: 8th International Symposium on Intelligent Data Analysis. Lyon 2009", S. 59-70, ISBN 978-3-642-03915-7
http://dx.doi.org/10.1007/978-3-642-03915-7_6

$$s(\vec{v_i}, \vec{v_j}) = \sum_{p,q=1}^{k_l} \left(|v_{ip} - \frac{1}{k_l}| \cdot |v_{jq} - \frac{1}{k_l}| \right) (i \neq j, p = q, 1 \leq (p,q) \leq k_l)) \qquad (3)$$

A high value from $s(\vec{v_i}, \vec{v_j})$ indicates a high significance of the determined *pa*-case. Now, we are able to determine a *pa*-case and an additional significance value for any pair of objects. To simplify matters we combine $s(\vec{v_i}, \vec{v_j})$ and $case()$ into one single function: $case^+(x_i, x_j) = case(x_i, x_j) \cdot s(\vec{v_i}, \vec{v_j})$. With this, the result interpretation changes slightly. Now, the determined *pa*-case is denoted by the algebraic sign of the result, while its absolute value represents the confidence of the decision. For undecidable *pa*-cases the function simply yields 0.

Regarding significance, at this point, we can only evaluate it in relation to other significance values, stating e.g that a_+ for $x_1; x_2$ has a higher significance than for $x_1; x_3$. To make assumptions about the significance on an absolute scale, we need to normalize our results, so that $case^+(x_i, x_j)$ yields 1 if a_+ holds and -1 if a_- holds with maximum significance. Therefore, we require the results of $case^+$ for the mentioned cases. We will illustrate this normalization with some examples, beginning with the most significant a_+ case. An example for this case, in a C_l with $k_l = 3$, would be given for $x_i; x_j$ with $\vec{v_i}^\top = \vec{v_j}^\top = (1, 0, 0)$. As simplification we assume for these examples that $v_{ip} = 1$ and $v_{ip} = 0$ can occur. Actually the strict definition for soft cluster assignments demands $\forall v_{ip} | 0 < v_{ip} < 1$. In this example, the most significant a_+ leads to $case^+ = \frac{2}{3}$. Using this setting, the most significant a_- occurs e.g. for $\vec{v_i}^\top = (1, 0, 0)$ and $\vec{v_i}^\top = (0, 0, 1)$ and results in $case^+ = -\frac{5}{9}$. We can see that the absolute values differ for both maximum significance cases. The reason for this behavior is $s(\vec{v_i}, \vec{v_j})$. It measures the distance from the fully balanced assignment in each dimension. We already know that $0 < v_{ip} < 1$, by $\frac{1}{k_l}$ this range is divided into two intervals. These have equal sizes for $k_l = 2$ but become disproportionate as k_l increases resp. $\frac{1}{k_l}$ decreases. This means that $v_{ip} > \frac{1}{k_l}$ can have a higher maximum distance to $\frac{1}{k_l}$ than $v_{ip} < \frac{1}{k_l}$. Based on this we define a norm considering k_l and integrate it into our $case^+$ method, thus creating our final function $case^{\|+\|}$:

$$case^{\|+\|}(x_i, x_j) = \frac{case^+(x_i, x_j)}{\|k_l\|}; \quad \|k_l\| = \begin{cases} 1 - \frac{1}{k_l} & \text{if case}(x_i, x_j) = 1 \\ -\frac{4}{k_l^2} + \frac{3}{k_l} & \text{if case}(x_i, x_j) = -1 \\ 1 & \text{if case}(x_i, x_j) = 0 \end{cases} \qquad (4)$$

4 Flexible Clustering Aggregation

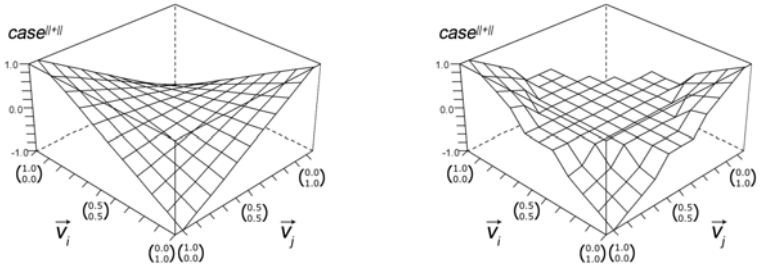
In this section, we describe how the expansions introduced in the previous section are integrated into the clustering aggregation to make it flexible and enable result adjustments. For the basic aggregation procedure, we adopt the idea described by Gionis et al. in [5]. Using our function $case^{\|+\|}$, we determine the *pa*-case for every object pair in all clusterings of $\mathcal{C}$. When deciding on the assignment case for $x_i; x_j$ in the aggregated result $\hat{C}$, we enact a majority decision and choose the

Final edited form was published in "Advances in Intelligent Data Analysis VIII: 8th International Symposium on Intelligent Data Analysis. Lyon 2009", S. 59-70, ISBN 978-3-642-03915-7
http://dx.doi.org/10.1007/978-3-642-03915-7_6

pa-case occuring the most for $x_i; x_j$. If no majority can be identified, e.g. if all three *pa*-cases have equal occurences, we decide for $a_?$ for the corresponding pair in the aggregate, since the final/global assignment is effectively undecidable. With this method, we can construct an aggregate but we are still lacking flexibility resp. control.

To achieve this control, we utilize the significance information provided by $case^{\|+\|}$ and filter all *pa*-cases according to their significance. This can be done with a filtering function that returns 0 if $case^{\|+\|}(x_i, x_j)| \leq t$ and $case^{\|+\|}(x_i, x_j)$ otherwise. The threshold t specifies the minimum amount of significance a pairwise assignment needs to have to be considered as decidable. Therefore, all assignments not exceeding t are classified as $a_?$ with zero significance. With this we are able to create an area of undecidability that allows us to mark not only balanced/fully balanced assignments as $a_?$, but also those assignments in their close proximity. Lets regard our example in Fig. 2 again: undecidable assignments are located on the grey lines and for pair $x_1; x_3$, a_+ holds with low significance. If we apply filtering, the grey lines of Fig. 2 expand and form an area of undecidability that can be described as a union of circles centered at intersection points and broadened lines/stripes. With increasing t the circles radii and width of stripes also increase. If the t-defined area is big enough to enclose x_3, its assignment becomes undecidable. Under these conditions, the pair $x_1; x_3$ is classified as $a_?$. Basically, via filtering we guarantee a minimal confidence for all decidable *pa*-cases.

In Fig. 3(a), the results of our function $case^{\|+\|}$ for the experimental setting from Section 2 are shown. We can observe our desired behavior of absolute and maximal significance scores at the plane corners. Take for example the western corner at $\vec{v_i}^\top = (1, 0)$ and $\vec{v_j}^\top = (1, 0)$, the *pa-case* for this pair is a_+ with maximum significance, so $case^{\|+\|}$ yields 1 at this point. The significance drops linearly towards and equals zero at the planes middle and its bisectors. The middle of the plane is specified by $\vec{v_i}^\top = (0.5, 0.5)$ and $\vec{v_j}^\top = (0.5, 0.5)$. This pair is composed of two objects with fully balanced assignments, making it undecidable i.e. $case^{\|+\|}$ yields zero. When we apply *filter* with threshold $t = 0.3$, the results change to Fig. 3(b). A flat area has formed around the center of the plane and



(a) $t = 0$ (b) $t = 0.3$

Fig. 3. Results of $case^{\|+\|}$ with and without filtering

Final edited form was published in "Advances in Intelligent Data Analysis VIII: 8th International Symposium on Intelligent Data Analysis. Lyon 2009", S. 59-70, ISBN 978-3-642-03915-7
http://dx.doi.org/10.1007/978-3-642-03915-7_6

its bisectors. None of the object pairs in this area satisfies the filtering criterion, and hence, is classified as $a_?$.

With the methods proposed so far, we are able to determine one of our three *pa*-cases on the aggregate level and can control the amount of $a_?$ via t. With this, we define stable cores in our aggregate–a_+,a_- robust against t–and around them, areas of undecidable $a_?$'s. These areas are the key to result flexibility and we introduce some examples of $a_?$-handling in the next section.

5 Evaluation

For the experiments in this section, we used a synthetic dataset consisting of 1500 objects. These objects form 7 clusters, where 2 clusters are very close but not linked and two cluster pairs are connected via bridges of different length and width. The dataset structure is depicted in Fig. 4. We used *k-means* [2] to generate our input clusterings. Due to the characteristics of our dataset and *k-means*, it is very unlikely that we obtain a good clustering using iterations with only single algorithm runs.

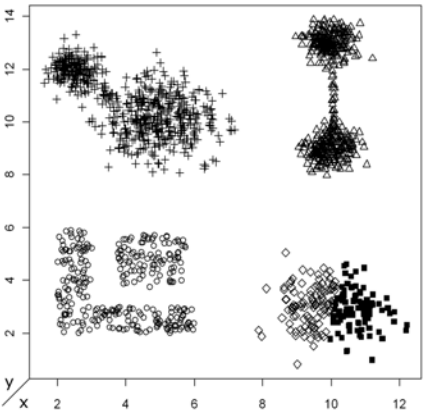
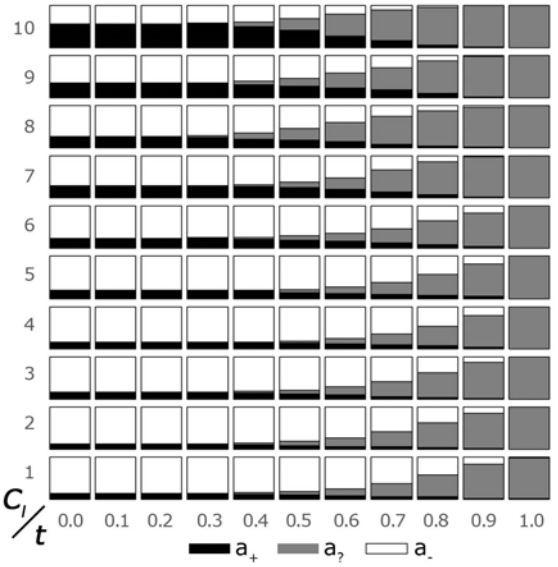


Fig. 4. $\hat{C}$ using scalar aggregation

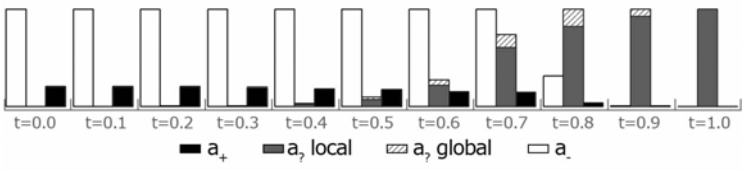
By applying existing aggregation approaches, we can already improve results, even for disadvantageous algorithm-dataset combinations. Therefore, we generate a $\mathcal{C}$ with 10 input clusterings using $k = \{2, 3, \cdots, 10, 15\}$ and different initializations that is aggregated using the technique described in [5]. Fig. 4 shows the obtained result, consisting of five clusters, where three clusters might be divided further while the remaining two clusters could be merged. We see that with clustering aggregation, a useful partitioning can be obtained even if singular algorithm execution yields suboptimal results. But this aggregation result is still not optimal and if the user wants to adjust it, he/she has to repeat the cycle: (i) modify parameters/algorithms of $\mathcal{C}$; (ii) recreate $\mathcal{C}$; (iii) execute aggregation; (iv) evaluate $\hat{C}$ until the desired adjustments occur.

For our flexible clustering aggregation, we use the same setup as before but change the algorithm to *FCM* [13], a soft version of *k-means*. Concerning the handling of $a_?$, we have to regard two alternatives, since we cannot determine if undecidable pairs are in the same cluster or not. Therefore, we define two strategies: one mapping $a_?$ to a_+ and another one that maps it to a_-. We let t run from zero to one in steps of 0.1, obtaining 11 aggregation results, one for each t. Fig. 5(a), shows the distribution of the determined *pa*-cases for all C_l and $\hat{C}$, monitored over all runs. Each block of the matrix displays the ratio of the *pa*-cases with reference to all object pairs of a C_l (specified by row) subjected to

Final edited form was published in "Advances in Intelligent Data Analysis VIII: 8th International Symposium on Intelligent Data Analysis. Lyon 2009", S. 59-70, ISBN 978-3-642-03915-7
http://dx.doi.org/10.1007/978-3-642-03915-7_6



(a) local *pa*-cases for $\mathcal{C}$



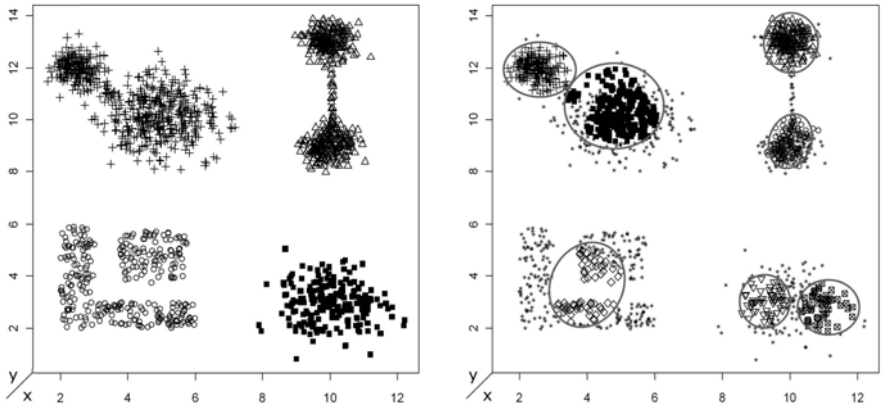
(b) global *pa*-cases for $\hat{C}$

Fig. 5. Evaluation results

filtering using t (specified by column). We observe that the number of $a_?$ rises with increasing t, whereas different C_l show different levels of robustness towards t. The distributions of the global *pa*-cases leading to $\hat{C}$ are shown in Fig. 5(b). We notice again that with increasing t the number of $a_?$ rises. In this diagram, $a_?$ *local* indicates $a_?$ as dominant, while $a_?$ *global* implies multiple dominant *pa*-cases and thus undecidability on the aggregate level. A major part of our future work will be the utilization of this significance information for the construction of $\mathcal{C}$ resp. evaluation of its clusterings.

We now adjust the aggregate by modifying t and $a_?$-handling, while $\mathcal{C}$ remains untouched. We choose $a_? \rightarrow a_+$ and increase t. With $t = 0.1$, we obtain the result shown in Fig. 6(a), where the two clusters in the lower right have been fused due to the points along the border between both former clusters. Having nearly equal affiliations to both clusters, they lead to *pa*-cases with low significance. Therefore, $a_?$ starts to occur near the border when $t = 0.1$ is applied. Since we map $a_?$ to a_+, both clusters are connected. If t increases further, more clusters connect leading to a unification of all datapoints at $t = 0.4$. This *merge* strategy

Final edited form was published in "Advances in Intelligent Data Analysis VIII: 8th International Symposium on Intelligent Data Analysis. Lyon 2009", S. 59-70, ISBN 978-3-642-03915-7
http://dx.doi.org/10.1007/978-3-642-03915-7_6



(a) Result for $t = 0.1$ and *merge* (b) Result for $t = 0.8$ and *split*

Fig. 6. Aggregation results

is very delicate since one single pair classified as a_+ is enough to merge whole clusters, that would otherwise be very dissimilar.

Next, we use $a_? \rightarrow a_-$, which yields the result shown in Fig. 4 at $t = 0$ and observe no changes in $\hat{C}$ until $t = 0.4$. At this point, an additional cluster forms and contains all objects the algorithm was unable to assign to a cluster because they are labeled $a_?$ and hence a_- in all of $\mathcal{C}$. Those objects are put into a *noise* cluster for convenience and presentation. Actually, each object is a singleton cluster for itself, since no affiliations to other objects or existing clusters can be determined, which is a novel trait that cannot occur in existing aggregation approaches. When we increase t, this *noise* grows, especially in areas equally influenced by multiple clusters. Fig. 6(b) shows the aggregation result for $t = 0.8$ with *noise* marked as *. We notice that the clusters in the upper quadrants were split by the *noise*. As $t \rightarrow 1$, all objects of the dataset become members of the *noise* cluster. During our experiments, we discovered that in contrast to our *merge* approach, this *split* strategy leads to slighter changes of the clustering aggregate. Part of our future work will deal with the construction of additional $a_?$-handling strategies as well as finding a method allowing independent selection of the handling strategy for each individual undecidable *pa*-case.

We showed that reasonable adjustments of $\hat{C}$ are possible using filtering and our proposed strategies *merge* and *split*. These adjustments can be easily made, since the required parameters can be abstracted to simple options. The handling strategies for $a_?$ effectively compare to "more clusters" for *split* and "fewer clusters" for *merge*, while t describes "how strong" each strategy is enforced. In summary, this section illustrated that clustering aggregation is beneficial even for algorithms not fitting the data. Futhermore, it described limitations of existing approaches and how to overcome them, using our flexible aggregation. Unfortunately, control and clustering flexibility come at the cost of runtime. Like all aggregation approaches utilizing pair-wise assignments, e.g. [5], our approach has

Final edited form was published in "Advances in Intelligent Data Analysis VIII: 8th International Symposium on Intelligent Data Analysis. Lyon 2009", S. 59-70, ISBN 978-3-642-03915-7
http://dx.doi.org/10.1007/978-3-642-03915-7_6

a complexity of $\mathcal{O}(n^2)$, with n being the number of data objects. Additionally, our approach use vector calculations that add to the runtime. Runtime optimization is a general field of research in the clustering area and a major part of our future research in particular but not focus of this paper.

6 Conclusion

In this paper, we proposed our flexible clustering aggregation approach that allows the construction of a clustering aggregate from a set of separate soft clustering results. We described the challenges of pair-wise assignments and decidability in this scenario and introduced (i) novel tools like the $a_?$ pair-wise assignments to master these challenges, (ii) a significance measure for *pa*-cases as well as (iii) a controllable aggregation process. All this enables our approach to produce adjustable results. We also simplified and abstracted our proposed parameters, thus allowing user-friendly and -guided identification of structures hidden from existing aggregation methods.

References

1. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proc. of KDD (1996)
2. Forgy, E.W.: Cluster analysis of multivariate data: Efficiency versus interpretability of classification. Biometrics 21 (1965)
3. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. ACM Comput. Surv. 31 (1999)
4. Zeng, Y., Tang, J., Garcia-Frias, J., Gao, G.R.: An adaptive meta-clustering approach: Combining the information from different clustering results. In: Proc. of CSB (2002)
5. Gionis, A., Mannila, H., Tsaparas, P.: Clustering aggregation. In: Proc. of ICDE (2005)
6. Boulis, C., Ostendorf, M.: Combining multiple clustering systems. In: Boulicaut, J.-F., Esposito, F., Giannotti, F., Pedreschi, D. (eds.) PKDD 2004. LNCS (LNAI), vol. 3202, pp. 63–74. Springer, Heidelberg (2004)
7. Strehl, A., Ghosh, J.: Cluster ensembles — a knowledge reuse framework for combining multiple partitions. Journal of Machine Learning Research 3 (2002)
8. Filkov, V., Skiena, S.S.: Heterogeneous data integration with the consensus clustering formalism. In: Rahm, E. (ed.) DILS 2004. LNCS (LNBI), vol. 2994, pp. 110–123. Springer, Heidelberg (2004)
9. Fred, A.L.N., Jain, A.K.: Robust data clustering. In: Proc. of CVPR (2003)
10. 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)
11. Long, B., Zhang, Z.M., Yu, P.S.: Combining multiple clusterings by soft correspondence. In: Proc. of ICDM (2005)
12. Topchy, A.P., Jain, A.K., Punch, W.F.: Combining multiple weak clusterings. In: Proc. of ICDM (2003)
13. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum, New York (1981)
14. Habich, D., Wächter, T., Lehner, W., Pilarsky, C.: Two-phase clustering strategy for gene expression data sets. In: Proc. of SAC (2006)