Dirk Habich, Martin Hahmann, Wolfgang Lehner

Evolving Ensemble-Clustering to a Feedback-Driven Process

Erstveröffentlichung in / First published in:

IEEE International Conference on Data Mining Workshops. Sydney, 13.12.2010. IEEE, S. 401-408. ISBN 978-1-4244-9244-2

DOI: https://doi.org/10.1109/ICDMW.2010.136

Evolving Ensemble-Clustering to a Feedback-Driven Process

Martin Hahmann, Dirk Habich, Wolfgang Lehner
Database Technology Group
Dresden University of Technology
Dresden, Germany
[*martin.hahmann, dirk.habich, wolfgang.lehner*]*@tu-dresden.de*

***Abstract*—Data clustering is a highly used knowledge extraction technique and is applied in more and more application domains. Over the last years, a lot of algorithms have been proposed that are often complicated and/or tailored to specific scenarios. As a result, clustering has become a hardly accessible domain for non-expert users, who face major difficulties like algorithm selection and parameterization. To overcome this issue, we develop a novel feedback-driven clustering process using a new perspective of clustering. By substituting parameterization with user-friendly feedback and providing support for result interpretation, clustering becomes accessible and allows the step-by-step construction of a satisfying result through iterative refinement.**

***Keywords*-ensemble-clustering; feedback; visualization**

I. Introduction

Clustering is described as the problem of partitioning a set of objects into groups, called clusters, so that objects in the same cluster are similar, while objects in different clusters are dissimilar [1]. Despite the multitude of available algorithms, data clustering remains a very challenging task, especially from the end user's perspective. The core problem is to select an optimal algorithm and parameterization for the particular task at hand, whereas it is not guaranteed that either of them exists[1], [2], [3]. So, in practice, the user has to complete the three steps: (i) algorithm selection, (ii) algorithm parameterization and execution, and (iii) result interpretation. If the result is interpreted as not satisfying, it is completely discarded. Thus, the three-step cycle is repeated over and over again, until the result is finally accepted. During these iterations, algorithms and/or parameter values are varied, for which the user has to rely on experience and intuition, since support for these actions is scarce. All this makes the derivation of a user-satisfying clustering result tedious work, since the unassessable number of necessary iterations and the perpetual re-computation of clusterings for the data waste a lot of time and resources.

In addition to the single-algorithm clustering described above, ensemble-clustering has established itself as an alternative clustering approach [4], [5]. This concept generates multiple clusterings for a dataset—the cluster-ensemble—and aggregates them to construct a final result. This procedure offers several benefits: aggregation results are more robust and show higher quality when compared to single-algorithm clustering [4], [5], [6], while algorithm selection and parameterization are eased since the focusing on one optimal algorithm-parameter combination is relaxed. Despite these differences, the overall process of ensemble-clustering still remains unchanged: algorithms and parameters are selected, the cluster-ensemble is generated, and the aggregation result is interpreted. Again, unsatisfying results are completely discarded and the process cycle is repeated with a modified cluster-ensemble.

In our opinion, the adherence to this established process restricts the opportunities of the ensemble-clustering concept. Therefore, we want to evolve this concept and describe a novel feedback-driven ensemble-clustering process. Our novel process exploits the capabilities of ensemble-clustering better, to support and involve the user during analysis. Moreover, it allows the construction of satisfying results through guided iterative refinement.

Our contributions are presented as follows: In Sec. II, we describe our modified perspective of clustering to which we align our process. Subsequently, the employed algorithmic platform is introduced in Sec. III, which is based on our ensemble-clustering approach [3]. Using this platform as a foundation, we define the necessary components of our process in Sec. IV: First, we derive the feedback necessary to navigate through our process from our aggregation method [3], which features an intuitive effect-oriented parameterization(Sec. IV-A). To enable the user to interpret obtained clustering solutions, we propose specific visual resources for that purpose in Sec. IV-B. The paper concludes with a short summary. Due to space limitation, evaluation and related work can be found in an extended version [7].

II. Our Perspective Of Clustering

A multitude of clustering algorithms already exist, of which many claim to generate clusterings with a higher quality than other methods. At the moment, result quality probably is the major focus and evaluation criteria in the development of clustering algorithms. While this orientation is obvious and reasonable at first sight, it shows a substantial problem: *There is no universally valid definition for clustering quality.* Many methods for clustering validation have been developed, of which we will name a few examples in the following. An often used approach is the comparison of a clustering result with a known optimal solution. If the optimal clustering solution is not known—which is the regular case—statistical tests based on the *null hypothesis* can be employed[8]. In addition, a wide range of indices measuring compactness and/or separation of clusters exist, e.g., *Dunn's index* or the *Davies-Bouldin index*[8]. Another way is relative validation, where the best result is chosen from a set of generated clusterings according to a pre-specified criterion[8].

While validation techniques can be very different, they share the one characteristic that their significance depends on the data that is clustered, the employed algorithm(-class) and its parameters. Thus, the selection of an appropriate validation method can be effectively considered as an additional parameter of cluster analysis.

In this paper, we introduce a different perspective to the problem; thus we evade some of the mentioned issues. When interpreting/evaluating the clustering result, we do not regard result quality. Using our obtained clustering scheme, we evaluate how "good" the underlying dataset ***fits*** to the scheme according to the constraints dictated by the applied algorithm(-class). In other words, we look at how "good" the data was clustered from the clustering algorithm's point of view. At this point, the concept of *"fit"* may seem vague and generic. Therefore, we provide more detailed explanations in the following sections and show the benefits of our perspective.

III. Algorithmic Platform

In this section, we introduce our ensemble clustering approach—*Flexible Clustering Aggregation (FCA)* [3]. This approach represents the algorithmic platform of our feedback-driven ensemble-clustering process.

The basic concept of FCA is clustering aggregation, which combines different clusterings of a dataset into one result to increase quality and robustness [3], [4]. Different aggregation approaches are known, where the pairwise assignment approach is considered as the most capable one. This approach evaluates each object pair of a dataset, determining whether it is assigned (i) to the same cluster or (ii) to different clusters. The aggregate is constructed by selecting the most frequent of these two pairwise assignments for each object pair and setting it in the result clustering. All existing aggregation techniques lack controllability, thus an aggregation result can only be adjusted through modification and re-computation of the input clusterings.

Our *Flexible Clustering Aggregation (FCA)* [3] tackles this issue. The key approach of our technique is to change the aggregation input from hard to soft clusterings [9]. These assign to each object its relative degree of similarity with all clusters instead of a hard assignment to just one cluster. Such assignments can be (i)generated by specific algorithms like *fuzzy c-means* [9] or (ii) calculated from arbitrary clustering results, using refinement techniques like *a-posteriori* [6]. In this paper, we will only use *fuzzy c-means* to generate the clusterings for our ensembles.

In a soft clustering, each datapoint $x_i|(1 \leq i \leq n)$ of a dataset $\mathcal{D}$ is assigned to all k clusters $c_j|(1 \leq j \leq k)$ of a clustering C to a certain degree. Thus, the assignment information of x_i in C is denoted as a vector $\vec{v_i}$ with the components $v_{ip}(1 \leq p \leq k)|0 < v_{ip} < 1$ and $\sum_{p=1}^{k} v_{ip} = 1$ describing the relation between x_i and the p-th cluster of C. This fine-grained information allows, e.g., the identification of undecidable cluster assignments given when objects have identical maximal similarities with multiple clusters. Assume a clustering with $k = 3$, and an object x_i with $\vec{v_i}^\top = (0.4, 0.4, 0.2)$. Using this assignment, we cannot decide whether x_i belongs to c_1 or c_2, although c_3 can be excluded. Based on this, it is easy to see that the worst case regarding decidability is given for assignments with $\forall v_{ip}(1 \leq p \leq k) = 1/k$, since they do not even allow the exclusion of clusters when it comes to clear cluster affiliations. Of these two kinds of undecidable assignments, we name the first *balanced* and the second *fully balanced* [3].

To incorporate this additional information, we expanded the pairwise assignment cases for the aggregation by adding an undecidable case that is valid for object pairs containing undecidable assignments. Furthermore, we derived a significance measure for pairwise assignments on that basis. This measure incorporates the intra-pair similarity of soft assignments and their decidability. The lower bound for decidability is defined as 0 or as an impossible decision and is given for the mentioned undecidable cluster assignments. The upper bound of 1 is given for objects with a single degree of similarity v_{ip} approaching 1 while all others approach 0. Basically, decidability shows the distance of $\vec{v_i}$ to the *fully balanced* assignment.

With this significance score, pairwise assignments are filtered and classified as undecidable if they do not exceed a certain significance threshold. Aggregation control or result adjustment, respectively, is exercised by this filtering and the handling of undecidable pairwise assignments during aggregation. Since *undecidable* is no valid option for a final object assignment, two handling strategies exist: one assumes that undecidable pairs are part of the same cluster, while the other assumes the opposite. These strategies and the filtering threshold act as parameters, allowing the merging or splitting of clusters without modifying the input clusterings[3]. In general, the relation between parameters and the clustering result is one of cause and effect. Parameters like k for k-means or ε for DBSCAN *cause* different *effects* in the clustering result, e.g., the fusion of clusters or changes in their size. To achieve a certain effect, it is crucial to know its associated cause, which is quite challenging. The FCA method overcomes this by allowing the direct specification of desired effects, namely: *merge* for fewer clusters or *split* for more clusters. In our original work [3], these effects

Final edited form was published in "IEEE International Conference on Data Mining Workshops. Sydney 2010",S. 401-408, ISBN 978-1-4244-9244-2
https://doi.org/10.1109/ICDMW.2010.136

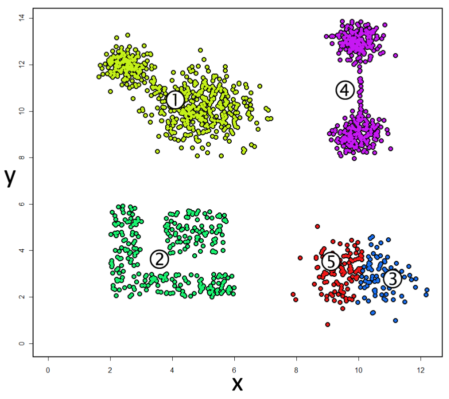


Figure 1. Toy Example

could only be applied to the whole clustering and were thus mutually exclusive. For this paper, we modified the algorithm so that effects can be applied to individual clusters.

IV. The Feedback-Driven Ensemble-Clustering Process

In this section, we describe the idea of our feedback-driven ensemble-clustering process in detail and introduce the components necessary for its realisation. The main idea of our process is to present an initial clustering of the data to the user and enable him to optimize it in an iterative fashion by giving feedback to the process. The initial clustering is generated by our algorithmic platform in the form of a clustering aggregate, which provides an advantageous starting point due to the robustness of aggregation results. As criterion for optimization we use the concept of *fit* as outlined in Section II. In order to execute our feedback-driven ensemble-clustering process, we have to develop a way to communicate the *fit* between the data and the current clustering schema, define the feedback itself and describe how feedback should be used to optimze the clustering result. During the following discussion of these three topics, we use the toy example depicted in Fig. 1 for exemplification. This initial clustering result for a synthetic dataset was generated using FCA without adjustments.

A. Defining Feedback

Our algorithmic platform offers two reasonable options for feedback realisation: (i) parameterization of the cluster ensemble and (ii) parameterization of FCA. The first option is discarded for the following reasons: On the one hand, the user needs to determine multiple sets of algorithm-specific technical parameters; on the other hand, the size of the ensemble and its composition must be specified, which is especially complicated if different clustering algorithms are employed in one ensemble. In contrast, the second option features a single point of parameterization and user-friendly *effect* parameters. Thereby, FCA can act as a kind of abstraction layer, since its parameters are independent from the algorithms used in the ensemble. On the algorithm level, FCA has two parameters: the handling strategy (*effect*) for undecidable pairs and the significance threshold. Based on these, we define the feedback for our ensemble-clustering process as a set of *feedback operations*.

A feedback operation normally consists of: the effect a user wants to achieve in the clustering and the *intensity* with which this effect should occur. The intensity matches the threshold for the significance filtering. A higher intensity leads to more pairs being filtered and becoming undecidable, meaning that more pairs are affected by the selected handling strategy/effect. We define the following feedback operations:

1) merge: This operation fuses clusters, selected by the user, into a single new cluster. During FCA, objects with undecidable pairwise assignments are treated as if they belong to the same cluster for this operation.

2) split: Using this operation, a cluster can be split into multiple clusters. The resulting number of clusters depends on the underlying cluster ensemble and the applied intensity. During the FCA, objects with undecidable pairwise assignments are treated as if they belong to different clusters. A split operation can result in the emergence of noise. This noise consists of singleton clusters, containing objects whose assignments are so weak that, with the applied filtering, no relation to an existing cluster can be determined.

3) refine: This operation removes weakly assigned objects—possible outliers—from a cluster and classifies them as noise. For this operation, objects with undecidable pairwise assignments are treated as if they belong to different clusters during FCA.

4) restructure: Up to now, our proposed feedback operations have modified clusters. The restructure operation is special since it does not directly change the cluster it is applied to but its underlying cluster ensemble. In some cases, it may happen that a cluster cannot be adjusted as intended, because the underlying cluster ensemble does not permit it. An example for such a case is cluster 2 in our toy example (see Fig. 1). To tackle this issue, the restructure operation takes the objects of the selected cluster and generates a new cluster ensemble for this data-subset that builds a new basis for cluster adjustment.

In one iteration of our feedback-driven ensemble-clustering process, the user can assign one of these operations to each cluster and then trigger the next iteration, where the specified adjustments are implemented. In the following iteration, the adjusted clusters are evaluated and further adjustments are applied if necessary. In the case that an applied feedback operation did not have the desired effect, the user can execute an *undo* that reverts the respective cluster to the state before the feedback operation was applied.

B. Resources for Interpretation

For the application of feedback operations it is essential to decide whether a cluster needs adjustments or not and, if the former case is given, what adjustments are appropriate.

This decision is made by the user and takes place during the interpretation of the clustering result. Therefore, to enable the user to make decisions in the first place, it is necessary to provide resources for the interpretation of clustering results.

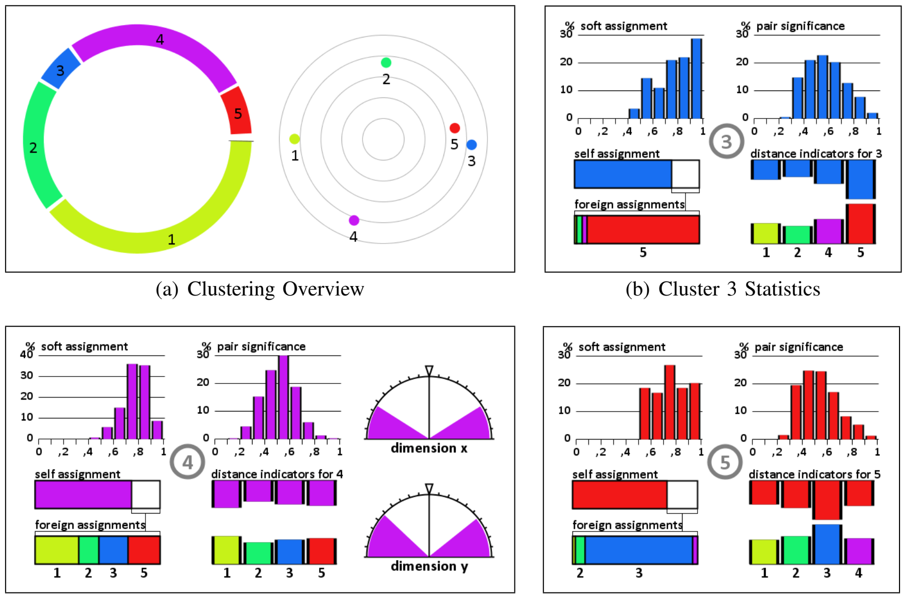
The most basic form of an interpretation resource is a scatterplot of the respective clustering. However, for datasets with high volume and high dimensionality this visualization is not sufficient. The validation measures mentioned in Sec. II form interpretation resources that can be applied to datasets of arbitrary scale. But due to the issues and our perspective on clustering described in Sec. II we do not use them.

Our goal is to optimize the clusters in reference to the algorithm(-class)-specified *fit* of clustering and dataset. Therefore, our resource for interpretation (*RFI*) must express how satisfying a certain cluster is, from the point of view of the employed clustering algorithm(-class). This makes it necessary to provide the user with algorithm(-class)-specific RFIs. Despite the mandatory adaption to certain algorithm-classes, we believe that a general template for RFIs can be derived from the abstract core objective of clustering. This objective is the partitioning of data into clusters, so that each cluster has *high internal similarity*, while all clusters can be *clearly separated* from each other. Although different algorithms implement these two conditions in different ways, they can be found in almost every method. Therefore, to specify an RFI, we need to include information about the *composition* of clusters and the *relations* between them. In our paper, we limit ourselves to the application of *fuzzy c-means*, thus, in the following, we describe the specific RFI constructed according to our template. To guarantee comprehensibility, we will construct our RFI using only simple visual components.

Composition: The fuzzy c-means is a partitioning algorithm, that assigns each data-object to the nearest of a set of predefined cluster prototypes(centroids). Therefore, a straightforward way to get information about the internal similarity of a cluster is to look at the distances between all cluster members and their respective centroid. In our scenario, this information is included in the soft assignments. All objects are assigned to the cluster to which they have the maximal soft assignment. These maxima are collected for each cluster and displayed as histogram. A second view on cluster composition is obtained by calculating the significance score for pairwise assignments (see Sec.III) for all object-pairs in a cluster. This information is again displayed as histogram. From the view of fuzzy c-means, a cluster has a high internal similarity if most of its objects/pairs are located near the centroid. For our RFI this means that the bulk of objects/pairs should be placed in the high assignment/significance bins of the two histograms. Examples are illustrated in Fig. 2(b),(c),(d) for our running example (upper part of the figures).

Additional information about the cluster composition are obtained by examining the distribution of the data in different dimensions. On the cluster level, it is desirable that all dimensions show unimodal data distributions (denoted as Φ), since this indicates intra-cluster homogeneity. In contrast, a multimodal distribution implies that the cluster could be further separated in the respective dimension. Therefore, we want to know the similarity of each cluster dimension to Φ. For this, histograms for each dimension of the dataset are generated per cluster and their local maximal bins are selected. Starting from each maximum, we iterate over the neighboring bins. If a bin contains a smaller or equal number of objects than the respective maximum, it is counted and the next bin is examined. This examination stops if the said condition is not fulfilled. With this method, we determine the maximum percentage of objects and bins (range) of a dimension that can be fitted under Φ. The higher this percentage, the more homogeneous is the cluster in this dimension. An example for this is shown in Fig. 2(c) (right part; dimensions x and y). The semicircle diagram shows the percentage of objects (left quadrant) and range (right quadrant) that *cannot* be fitted under Φ. More color in this scale shows less cluster homogeneity in the respective dimension. The example diagrams for Cluster 4 of our running example indicate that this cluster has a low internal similarity (see Fig. 1).

Relations: The most obvious way to express relations between clusters is their distance to each other. There are several approaches to measure the distances between groups/clusters of objects. Since we use fuzzy c-means, we compute the distance between centroids on the one hand and the minimal distance between the members of different clusters on the other hand. The centroids are only representatives of groups, so the second distance measurement provides more local details. The inter-centroid distances are displayed as a distance graph like the one depicted in the right part of Fig. 2(a), where each numbered circle represents a cluster centroid. Additionally, these centroid-to-centroid distances are combined with the minimal-object-distances and form the distance indicators shown in Fig. 2(b),(c),(d). Each pair of opposing columns illustrates the ratio of both mentioned distance types, between a fixed selected cluster and one of the remaining clusters. If such a pair of columns nearly ”touches”, both associated clusters also nearly ”touch”, meaning that they are not very well separated. The left part of Fig. 2(a) illustrates the number of clusters and the corresponding cluster sizes in a circle diagram. We already used soft assignments to get information about a cluster's composition. Since these assignments contain the similarity/relation of an object with all clusters, they are natural candidates for relation-based measures. For our RFI, we derive two measures from them: (i) self-assignment and (ii) foreign-assignment, which are defined as:



(a) Clustering Overview

(b) Cluster 3 Statistics

(c) Cluster 4 Statistics

(d) Cluster 5 Statistics

Figure 2. Information Source for the Initial Clustering Result

$$A_{self}(p) = \sum_{v_i \in c_p} v_{ip}; \qquad A_{foreign}(p,q) = \sum_{v_i \in c_p} v_{iq} | q \neq p \tag{1}$$

The self-assignment $A_{self}(p)$ of a cluster c_p sums up the degrees of similarity of all cluster members to the cluster itself. In contrast, the foreign-assignment $A_{foreign}(p,q)$ of a cluster c_p to a cluster c_q shows the influence c_q has on the objects of c_p. These sums are normalized to get a percental score. Both values are depicted as bars in the lower parts of Fig. 2(b),(c),(d). Reduced self-assignment of a cluster implies that other clusters are nearby and thus indicate an unclear separability of clusters. Actually, self-assignment has a hybrid character: since it is based on the assignments of cluster members and cluster representatives, it also provides information about cluster composition.

The measures described so far constitute our RFI for fuzzy c-means, allowing the user a visual evaluation of cluster conditions and the determination of appropriate feedback operations.

C. *Selecting appropriate Feedback*

With the RFI defined, the user is able to identify clusters that need adjustments and select appropriate feedback operations. In the following, we describe which feedback operation should be employed if certain cluster characteristics occur.

1) merge: Identification of this feedback operation is relatively easy. Clusters that should be merged show (i) close proximity in the inter-centroid distances and distance indicators, (ii) an accumulation of objects and pairs in the medium to lower bins of the soft assignment and pair significance histograms, and (iii) a reduced self-assignment in combination with a very strong foreign assignment towards the merge partner. If we look at Fig. 2(b) and (c), we observe that these properties hold, hence Cluster 3 and Cluster 5 of our running example should be merged.

2) split: In contrast to merge, split operations are harder to identify. A typical split candidate shows a clear separation from other clusters in distance indicators and in inter-centroid distances. It also features a reduced self-assignment, but its foreign assignments are mostly balanced over the remaining clusters. To decide whether or not a split should be applied, the dimension diagrams must be consulted additionally. If these show a significant percentage of objects/bins that cannot be fit under a unimodal distribution, a split should be applied. Figure 2(c) represents such a case and Cluster 4 should be split into several clusters.

3) refine: This operation should be applied to make final adjustments to nearly perfect fitting clusters. These feature a very high self-assignment, while nearly all of their objects and object pairs are located in the highest assignment or significance bins. In addition, the inter-centroid distances show that they are well separated from other centroids. With refine, outliers can be removed from these clusters. Outliers can manifest themselves as barely populated bins in the soft assignment and pair significance histograms. They can also lead to distance indicators, showing close proximity to a neighboring cluster even if all other measures contradict this.

4) restructure: This operation is applied to clusters where adjustments have failed, although all feedback operations were used to their full extent. A restructure can also be applied if the identification of a feedback operation is very ambiguous for a particular cluster. With this, it is possible to get a new starting point for the adjustment of the respective cluster that potentially allows clearer decision making.

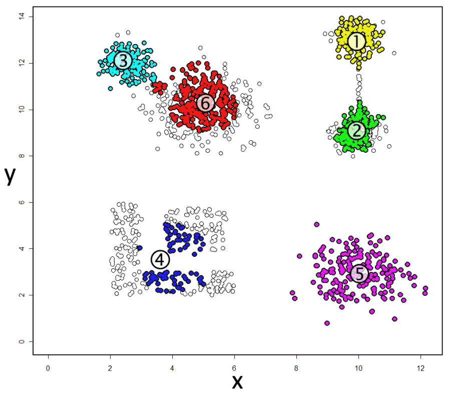


Figure 3. Clustering after the First Iteration

V. Process Application

All single components of our feedback-driven clustering approach have been introduced. Now, we are able to illustrate the complete process in the form of a hands-on report using our running example, which is given in this section. Since this example covers multiple iterations, we use the notation $c_{\text{id}}^{\text{iteration}}$ when refering to a certain cluster in a certain iteration.

Initial Clustering: Figure 1 depicts the clustering result at iteration 0 of our feedback-driven clustering. By visual interpretation of the scatterplot or the illustrated part of the RFI in Fig. 2, we determine that all clusters are poorly constructed according to the principles of the employed fuzzy c-means. That means the *fit* between clustering schema and data is not very good; therefore, adjustments to the schema are necessary. Using the RFI, we derive the following appropriate feedback options: split is chosen for c_1^0, c_2^0 and c_4^0 since these clusters have weak compositions, i.e., a low internal similarity. In addition, merge is applied for clusters c_3^0 and c_5^0. Both clusters have a strong relation to each other, thus violating the separation criteria. The intensity of each operation can be derived from the pair significance histogram. For merge operations, the intensity can be very low since transitive effects allow the fusion of clusters by few object pairs. For the merge of c_3^0 and c_5^0, we choose an intensity of 0.3 resembling the upper bound of the lowest populated pair significance bin. During previous experiments on FCA, we have discovered that split operations generally require a higher intensity; thus, for c_4^0, we set the intensity to 0.6.

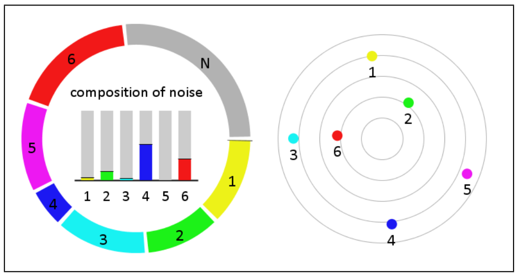
After the assignment of feedback operations, we start the first iteration of our ensemble-clustering process where the specified adjustments are realized. Feedback operations are assigned to individual clusters; we have to assure this locality during aggregation. The original FCA [3] always evaluates all pairs of the dataset and thus allows no cluster-specific aggregation and handling strategy. To overcome this, we extended FCA, so that for each feedback operation, only the object pairs in the cluster(s) linked to the particular operation are processed. For example, to merge c_3^0 and c_5^0, only object pairs from these clusters are evaluated by FCA. Using this approach, we can localize handling strategies in FCA and also reduce the number of pairs needed to be processed.

Iteration 1: The result of the first iteration is shown in Fig. 3. We observe that the merge and two split operations were successful. The split of c_4^0 results in two clusters c_1^1 and c_2^1, while the merge of c_3^0 and c_5^0 generated c_5^1. The statistics for c_1^1 and c_5^1 are illustrated in Fig. 4(b) and (d), showing high internal homogeneity as well as strong separation from other clusters. Therefore, both clusters now fit the data according to our definition.

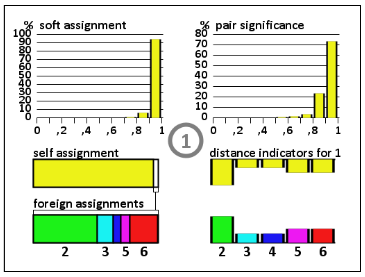
This also applies for the clusters c_2^1, c_3^1 and c_6^1. These clusters require no further adjustment, but a user may still apply the refine operation to remove possible outliers. In comparison to the initial clustering result, iteration 1 features noise. Noise is generated by split operations and actually consists of singleton clusters. As described, the split operation handles undecidable pairs as being assigned to different clusters. It can happen that all pairs a certain object is part of become undecidable. Thus, this object is not in a cluster with any other object of the dataset, which means that it forms a (singleton) cluster itself. The amount of noise generally corresponds to the intensity of the split. In our scatterplots, noise is depicted as non-colored dots.

Regarding Fig. 3, we observe that the small bridge that connected c_1^1 and c_2^1 in the initial clustering has turned into noise as well as objects at some cluster borders. This is also the case for c_4^1, which we examine in detail. The left part of Fig. 4(a) shows that more than a quarter of the dataset is classified as noise (marked by the segment labeled with N). The composition of noise is determined by calculating the closest centroid for each noise object, revealing that half of the noise is located in the proximity of c_4^1 and was thus part of the cluster in the previous iteration. From this and the statistics for c_4^1 depicted in Fig. 4(c), we derive that the split of c_4^0 failed to produce multiple new clusters and transformed a major part of its members into noise. The resulting c_4^1 is still well separated and shows a slightly increased soft- and self-assignment. But despite this apparent increase, pairwise significance is not maximized and dimension y still indicates a split. With this in mind, the success of a subsequent split with increased intensity seems unlikely, thus leaving the conclusion that this cluster cannot be fitted to the data, using the underlying example. Therefore, *undo* is used to revert the cluster to the state before split was applied. Afterwards the restructure operation is assigned to the restored cluster c_4^0 and executed, which leads to the next iteration.

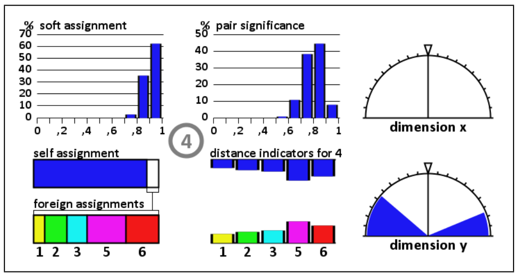
Iteration 2: During this iteration, the restructure operation generates a new cluster ensemble for c_4^0. For the new ensemble for this subset of data, a number of fuzzy c-means results are computed, where the cluster numbers range from 2 to 5. Subsequently, the aggregation result using FCA is computed and depicted in Fig. 5(a). We observe that



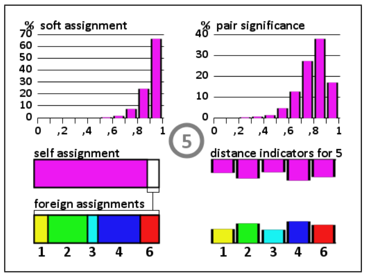
(a) Clustering Overview



(b) Cluster 1 Statistics



(c) Cluster 4 Statistics



(d) Cluster 5 Statistics

Figure 4. Information Source for the Clustering Result after Iteration 1

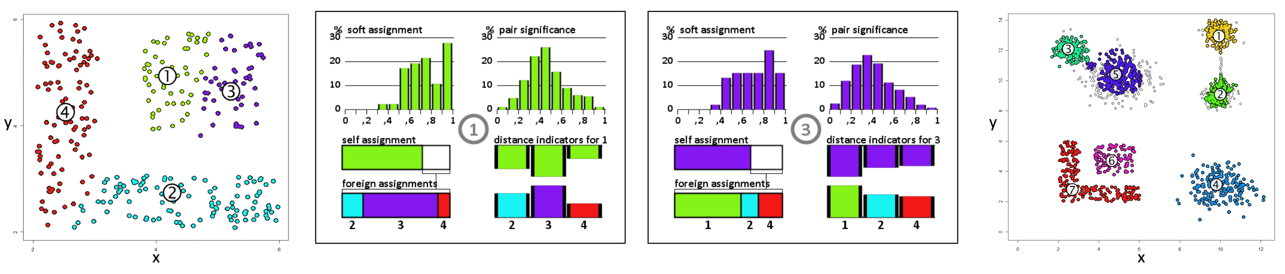
four clusters have formed that again do not *fit* the data. The RFI for c_1^2 and c_3^2—depicted in Fig. 5(b) and (c)—shows familiar characteristics indicating a merge operation for both clusters. Because the remaining clusters c_2^2 and c_4^2 show similar behavior, a merge operation is issued for them. The necessary intensities are again derived from the pair significance histograms.

Iteration 3: Both executed merge operations were successful and only two clusters remain. At this point, we decide that no further adjustments are necessary and re-integrate the generated clusters into the whole clustering. This reintegration represents iteration 4, whose result is depicted in Fig. 5(d). The plot shows that this result is quite the optimal clustering for this dataset. Restructuring and subsequent merge produced the clusters c_6^4 and c_7^4 that accurately segment the data in this part of the dataset. Looking at the scatterplot, we can visually determine that both clusters *fit* the data. In contrast, our RFI in Fig. 6 shows a rather unclear picture. Regarding the details of c_6^4 and c_7^4 (see Fig. 6), we observe reduced internal similarity and strong foreign assignments between both clusters, which would indicate a *merge*. But this assumption is contradicted by the firm separation shown by inter-centroid distances and distance indicators. From the user's perspective, this situation is ambiguous, since the clusters apparently do not *fit* the data and appropriate feedback cannot be cleary identified. The reason for this behaviour lies in the structure of the respective parts of the dataset. The clusters in this region are not spherical and can thus not be separated by centroid-based algorithms like fuzzy c-means, and exactly this is shown by our RFI. Basically, this means that this region of the dataset cannot be clustered effectively using fuzzy c-means. Based on this knowledge, there are two options to handle the situation.

The first one is compromise: The user has to evaluate the available options. An application of refine is obviously out of the question. Merging of both clusters would lead back to the cluster c_4^0 that clearly indicated a split, while further splitting of c_6^4 and c_7^4 would yield a result similar to iteration 2, where merge operations were explicitly identified. This only leaves another application of restructure, whereas an improvement of the situation is questionable since this operation has already been applied. A possible restructure of c_6^4 would, for example, most likely result in a number of clusters, showing clear merge characteristics. In summary, all feedback operations the user issues to get out of this unclear situation would lead to situations clearly indicating the return to the former unclear situation. So, in this setting, the user would accept the current result as best compromise even if the RFI contradicts this. Actually, this solution to the problem would be good in our example. The clusters in question show the optimal separation for the data, although the used algorithm could not provide this result without the benefits of our FCA.

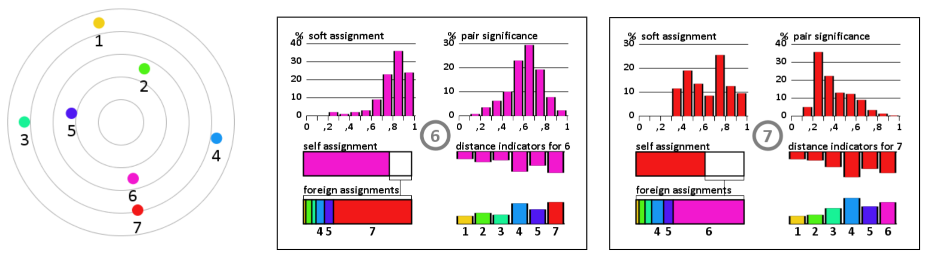
Nonetheless, the compromise is an impure solution regarding our *fit* paradigm. A consistent solution approach incorporates the reasoning: With the given feedback operations the clustering cannot be *fitted* to the data. Since the definition of *fit* is dictated by the employed algorithm through the RFI, this means that the respective area of the dataset cannot be clustered with the algorithm currently employed. This shows another benefit of our ensemble-clustering process. While feedback operations ease algorithm parameterization and RFI supports interpretation, the character of the *fit* paradigm provides information concerning the selection of clustering algorithms. Let us regard the running example:

Final edited form was published in "IEEE International Conference on Data Mining Workshops. Sydney 2010",S. 401-408, ISBN 978-1-4244-9244-2
https://doi.org/10.1109/ICDMW.2010.136



(a) Cluster 1 Statistics (b) Cluster 1 Statistics (c) Cluster 3 Statistics (d) Final Clustering

Figure 5. Information Source for the Restructured Cluster 4



(a) Overview (b) Cluster 6 Statistics (c) Cluster 7 Statistics

Figure 6. Information Source for the final Clustering

Facing the described ambiguous setting and after exploiting all feedback options, the user would change the employed algorithm(-class) for the respective region of the dataset. With that, *fit* can be achieved under the point of view of the new algorithm(-class). In this paper, we limited ourselves to the use of fuzzy c-means to describe a first version of our ensemble-clustering process. The mentioned change of algorithm(class) and its particular consequences will be part of our future research.

VI. Conclusion

In this paper, we introduced our novel feedback-driven ensemble-clustering process. In contrast to the established clustering procedure, that perpetually repeats algorithm execution and completely discards results that are not satisfying, our proposed process allows the iterative refinement of a clustering. This means that satisfying parts of the result are kept, while the remainder is adjusted. During the process flow, support is offered for all three necessary clustering steps: algorithm selection, algorithm execution and result interpretation. Our enhanced aggregation approach [3] was utilized to develop a compact set of easy-to-understand feedback operations that ease parameterization. By introducing the concept of *fit* between clustering and dataset we presented an alternative perspective and optimization criterion for clustering in general. Based on this concept we described resources for interpretation that support the user in the evaluation of an obtained clustering. In addition the concept of *fit* allows to determine whether an applied algorithm is appropriate for certain parts of the dataset or not, which can be used as advice regarding the selection of clustering algorithms.

References

[1] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Comput. Surv.*, vol. 31, no. 3, 1999.

[2] A. Jain and M. Law, "Data clustering: A users dilemma," *Pattern Recognition and Machine Intelligence*, pp. 1–10, 2005.

[3] M. Hahmann, P. Volk, F. Rosenthal, D. Habich, and W. Lehner, "How to control clustering results? flexible clustering aggregation," in *Advances in Intelligent Data Analysis VIII*, 2009, pp. 59–70.

[4] A. Gionis, H. Mannila, and P. Tsaparas, "Clustering aggregation," *TKDD*, vol. 1, no. 1, 2007.

[5] A. Strehl and J. Ghosh, "Cluster ensembles — a knowledge reuse framework for combining multiple partitions," *Journal of Machine Learning Research*, vol. 3, 2002.

[6] Y. Zeng, J. Tang, J. Garcia-Frias, and G. R. Gao, "An adaptive meta-clustering approach: Combining the information from different clustering results," in *Proc. of CSB*, 2002.

[7] M. Hahmann, D. Habich, and W. Lehner, "Evolving ensemble-clustering to a feedback-driven process - full version," 2010. [Online]. Available: http://wwwdb.inf.tu-dresden.de/files/publications/HHL+10.pdf

[8] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "On clustering validation techniques," *Journal of Intelligent Information Systems*, vol. 17, pp. 107–145, 2001.

[9] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York: Plenum, 1981.