Martin Hahmann, Dirk Habich, Wolfgang Lehner

Visual Decision Support for Ensemble Clustering

Erstveröffentlichung in / First published in:

Scientific and Statistical Database Management: 22nd International Conference. Heidelberg, 30.06. - 02.07.2010. Springer, S. 279-287. ISBN 978-3-642-13818-8.

DOI: http://dx.doi.org/10.1007/978-3-642-13818-8_21

Visual Decision Support for Ensemble Clustering

Martin Hahmann, Dirk Habich, and Wolfgang Lehner

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

Abstract. The continuing growth of data leads to major challenges for data clustering in scientific data management. Clustering algorithms must handle high data volumes/dimensionality, while users need assistance during their analyses. *Ensemble clustering* provides robust, high-quality results and eases the algorithm selection and parameterization. Drawbacks of available concepts are the lack of facilities for result adjustment and the missing support for result interpretation. To tackle these issues, we have already published an extended algorithm for ensemble clustering that uses soft clusterings. In this paper, we propose a novel visualization, tightly coupled to this algorithm, that provides assistance for result adjustments and allows the interpretation of clusterings for data sets of arbitrary size.

1 Introduction

Advanced scientific applications, e.g., gene expression analyses in biology and medical science, come with increased amounts of input data and performance requirements. This leads to two major challenges for data clustering. On the one hand, appropriate algorithms must be developed, while on the other hand, users require assistance with the application of such algorithms and the interpretation/perception of the vast data sets involved in the clustering process.

Fundamentally, clustering is defined as the problem of partitioning a set of objects into groups, so-called clusters [1,2], where objects in the same cluster are similar, while objects in different clusters are dissimilar. In order to create a clustering, the user has to complete three steps: (i) algorithm selection, (ii) algorithm execution (including parameterization), and (iii) result interpretation. Each of these steps has critical impact on the clustering process/result. The selection of the best-fitting clustering algorithm and parameters is a non-trivial task, additionally complicated by the multitude of available algorithms [2,3]. Therefore, the most common workflow for clustering in practice is found in the constant iteration over the mentioned three steps until a user-satisfying result is created. However, this iterative approach—corresponding to a 'trial and error' procedure—is tedious work and wastes time and resources.

To overcome this issue, *ensemble clustering* has been proposed [3,4,5,6]. This approach aggregates several partitionings of a data set—the cluster ensemble—into a final clustering result. The aggregate shows increased quality and robustness in comparison with the single input clusterings [3,4,5,6]. While the creation

Final edited form was published in "Scientific and Statistical Database Management: 22nd International Conference. Heidelberg 2010", S. 279-287, ISBN 978-3-642-13818-8
http://dx.doi.org/10.1007/978-3-642-13818-8_21

of a robust result is eased, existing methods lack instruments to enable the user to adjust/refine the obtained clustering result. Our algorithm proposed in [7] helps overcome this issue. A user is now able to modify a result, not by supplying 'technical' parameters but by choosing an intended *effect*, namely: *merge* if fewer clusters are desired or *split* if more clusters are needed.

For an effective refinement, the interpretation of the result and the derivation of corresponding adjustments are important. Our novel visualization concept proposed in this paper, which is tightly coupled with our extended ensemble clustering, assists the user in identifying the optimal *effects* for result refinement. In addition, our approach simplifies the interpretation of clustering results for data sets of arbitrary size. We start by introducing our ensemble-clustering scenario in Sec. 2. Subsequently, an in-depth description of our visualization is given in Sec. 3. In Sec. 4, we propose our software demonstration, before we end this paper with a conclusion and some remarks on future work.

2 Overview

The overall goal of our research project AEGIS is to enable users to conduct clustering processes in a simplified and efficient way, regardless of their background knowledge in the area of data mining. We want to achieve this goal by assisting the user during the mentioned three steps: (i) selection, (ii) execution, and (iii) interpretation. Our scenario is illustrated in Fig. 1, where two domains face each other. The *user domain* on top contains the data that is to be analyzed, the user's context knowledge about the data, and the clustering result. The *algorithm domain* at the bottom incorporates all existing algorithms for the clustering analysis, including their parameters. At the center, we find the three-step clustering cycle, during which the input data is passed to the algorithm domain for analysis, from where the clustering result is returned to the user domain for interpretation. The separated domains exemplify the average user's lack of knowledge in the area of algorithms and parameters. We have already introduced the ensemble-clustering concept as an effective way to support the

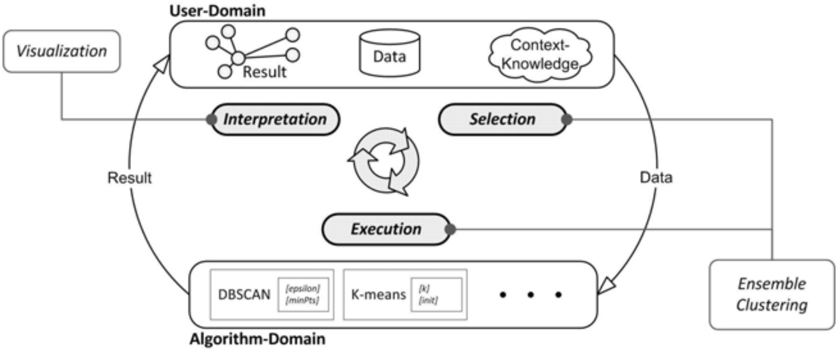

Fig. 1. The clustering scenario

user during algorithm selection and execution. Currently, most aggregation algorithms use a set of hard clustering results as basic input. Such hard clusterings are, e.g., obtained with k-means or DBSCAN, and they assign each object exclusively to the one cluster with the highest similarity to it. Based on this mapping, a pairwise assignment is determined for each object pair in every input clustering. Two cases of pairwise assignments are known: a pair of objects can either be located (i) in the same cluster or (ii) in different clusters. Using these assignments, the final aggregate is constructed by selecting the most frequent of both cases for each object pair and setting it in the aggregation result.

Influence of Input. The major drawback of all existing aggregation approaches is their lack of controllability. Suppose an aggregate does not satisfy the user, then the user's only option for result adjustment is the modification of the input clusterings. In this case, the actual benefits regarding user support are lost, since algorithm and parameter adjustments now need to be made for a whole set of clusterings. In addition, the modified input must be re-computed, which costs time and resources, especially for large cluster ensembles. With the enhanced aggregation concept we proposed in [7], both of these issues are tackled. The key of our technique is to change the aggregation input from hard to soft clusterings. Such clusterings assign to each object its relative degree of similarity with all clusters. They can be obtained via algorithms like *FCM* [8] or via refinement techniques like the a-posteriori approach [4]. The major benefit of soft assignments is the fine-grained information about object-cluster relations they provide. This allows, for example, the identification of undecidable cluster assignments that are found if an object has the same maximal similarity to more than one cluster. This may occur in every clustering but cannot be handled by hard assignments; it is thus ignored or randomly solved, respectively.

Flexibility of Aggregation. To utilize soft clusterings to the full extent, we expand pairwise assignment cases by adding an undecidable case that represents object pairs with undecidable assignments. Based on this, we propose a significance measure for pairwise assignments, including the intra-pair similarity and decidability for the respective object assignments. The range for decidability is defined as follows: zero decidability (meaning a decision is impossible) is given to objects with a maximal degree of similarity with more than one cluster. The maximum decidability is given if one of the object's degrees of similarity approaches 1 while all others approach 0. The decidability values for all objects are inside this range.

Based on our significance score, we define a filter that classifies all pairwise assignments not exceeding a certain significance threshold as undecidable. With this, actual aggregation control becomes possible. The mentioned result adjustments are achieved via such control and the handling of the undecidable pairwise assignments during aggregate construction. Since *undecidable* is no valid option for a final object assignment, we propose two handling strategies: one assumes that undecidable pairs are part of the same cluster, while the other assumes just the opposite. Our evaluation shows that these two strategies allow to merge or split clusters without modifying the original cluster ensemble, thus saving time and resources. Now, one could argue that there are still parameters burdening the user, which would normally be true, but our control parameters have a novel character. In general, the relation between parameters and the clustering result is one of cause and effect. With common 'technical' parameters like k for k-means or ε for DBSCAN, the user modifies the *cause* and awaits the effect regarding the cluster number and size. In our approach, the user simply chooses the desired *effect*, namely: *merge* for fewer clusters or *split* for more clusters.

Until now, merging and splitting have been mutually exclusive and had to be set for the whole clustering. This is sufficient if the bulk of clusters requires the same operation, but it effectively prevents an individual handling of clusters. In tight coupling with our extended aggregation approach, our developed visualization shall enable the user to interpret the obtained clustering and assist in the decision on whether or not clusters are stable and should be merged or split. With this, the result quality can be iteratively refined, whereas the provided support keeps the iteration count low.

3 Augur Visualization

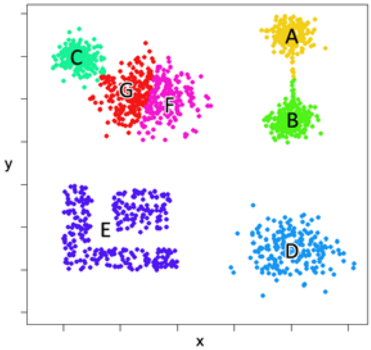

Fig. 2. Example aggregate

This section introduces our visualization by describing its input, its single views, the information visualized, and its interpretation. The input consists of the clustering aggregate provided by our algorithm [7], offering access to cluster centroids and sizes, soft cluster assignments, and significance scores for object pairs. From this input, additional information is computed for certain views, which will be explained during the description of the respective view. On the basis of Shneiderman's mantra, *'overview first, zoom and filter, then details-on-demand'* [9], our visualization features three views: overview, cluster composition and relations, and the attribute view. With this, we want to enable the user to determine the clusters that need no adjustment and to decide which ones should be merged or split, with the goal to improve the quality of the result. In this section, the clustering aggregate depicted in Fig. 2 is used as an example. It has been generated with ensemble clustering and its already good partitioning still needs adjustments. In all figures, clusters are identified via color.

Overview. The overview is the first view presented to the user and depicted in Fig. 3. This view is completely result-driven, i.e., only characteristics of the clustering aggregate are shown. The dominant circle represents the clusters of the aggregate, whereas each circle segment corresponds to a cluster whose percental size correlates with the segment's size. The radar-like gauge located on the left shows

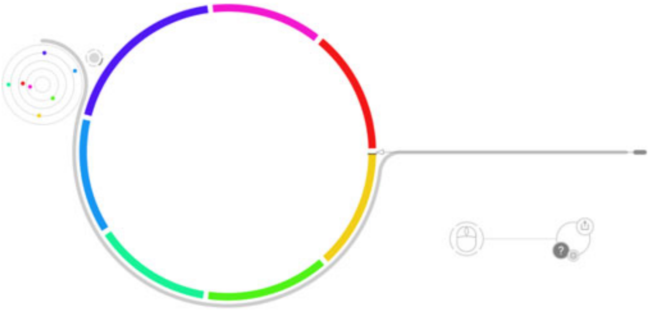

Fig. 3. AUGUR overview showing clusters and inter-cluster distances

the distances between the prototypes (centroids) of all clusters. The mapping between centroids in the radar and circle segment is done via color. The radar shows a distance graph, where vertices represent centroids, and edges—invisible in our visualization—represent the Euclidean distance between centroids in the full dimensional data space. Therefore, the radar is applicable for high-dimensional data. Since all our views are basically result-driven, we can also handle high-volume datasets without problems. The overview provides the user with a visual summary of the clustering result, allowing a first evaluation of the number of clusters and relations between clusters expressed by distance and size.

Cluster Composition and Relations. If the user identifies clusters of interest in the overview, e.g., two very close clusters like the pink (F) and red (G) ones in Fig. 2, they can be selected individually to get more information about them, thus performing *'zoom and filter'*. Cluster selection is done by rotation of the main circle. As soon as a cluster is selected, the composition and relations (c&r) view depicted in Fig. 4 (for cluster F) is displayed. The selected cluster's composition is shown by the row of histograms on the right. All histograms feature the interval $[0, 1]$ with ten bins of equal width. From the left to the right, they show the distribution of: (i) fuzzy assignment values, (ii) significance scores for all object-centroid pairs, and (iii) significance scores for all object-object pairs in the selected cluster. For details concerning these scores, refer to [7]. Certain histogram signatures indicate certain cluster states, e.g., a stable and compact cluster is given if all three histograms show a unimodal distribution with the mode—ideally containing all objects—situated in the right-most (highest significance) bin.

Let us regard the signature of the example depicted in Fig. 4. The histograms show that many of the object-centroid and pairwise assignments are not very strong. This indicates that there are other clusters (G in the example) that strongly influence the selected cluster objects, which leaves the chance that these clusters could be merged. To support such assumptions, the relations between clusters have to be analyzed. For this, the two 'pie-chart' gauges and arcs inside the main circle are used. The smaller gauge shows the degree of 'self-assignment' of the selected cluster, while the other one displays the degree of 'shared assignment' and its distribution among the remaining clusters. These degrees are

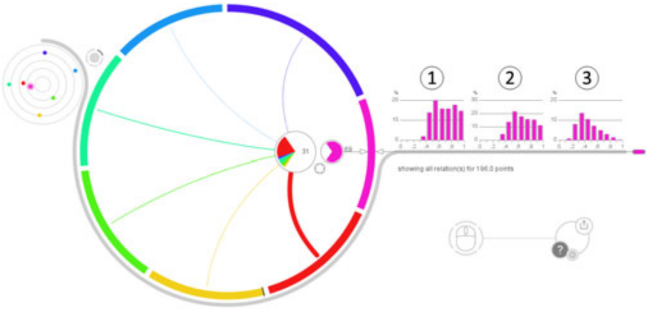

Fig. 4. AUGUR c&r view showing composition and relations for the pink cluster

calculated as follows: each fuzzy object assignment is a vector with a sum of 1, consisting of components ranged between 0 and 1, indicating the relative degree of assignment to a certain cluster, i.e., each vector-dimension corresponds to a cluster. The degree of self-assignment is calculated by summing up all components in the dimension corresponding to the selected cluster. This sum is then normalized and multiplied with 100 to get a percental score. The shared assignment is generated in the same fashion for each remaining cluster/dimension. The target and strength of relations between the selected cluster and others is described by the color and size of the shared-assignment slices. For easy identification, the displayed arcs show these cluster-to-cluster relations by connecting clusters, where the stroke width shows the strength of the relation.

If a cluster is not influenced by others, it shows a very high degree of self-assignment with no outstanding relations to other clusters. In contrast, the example in Fig. 4 shows that the selected cluster has a noticeable relation to the red cluster. This supports the merge assumption and furthermore indicates which other cluster should be part of a possible merge. To get additional information, the inter-cluster distances can be analyzed. For this, the user can employ the 'radar', showing that both clusters in our example are relatively close to each other (the selected cluster is encircled), or switch on additional distance indicators (*'details-on-demand'*), as shown in Fig. 5. These display the ratio of centroid-to-centroid distances—like the radar—and minimum object-to-object distances between the selected and the remaining clusters. If this ratio approaches 1, the respective clusters are well separated and the colored bars are distant. In our example, this is the case for all clusters except for the red one, where both bars nearly touch each other, showing that the minimal object distance between the clusters is much smaller than the centroid distance. With this, the user can now savely state that the pink and the red cluster should be merged. To double-check, the red cluster can be selected and should show similar relations to the pink one.

With the c&r view, it is also possible to evaluate whether or not a cluster should be split. Candidates for a split show the following: In all three histograms, the mode of the distribution is located in one of the medium-significance bins. Additionally, they feature a reduced degree of self-assignment, but in contrast to

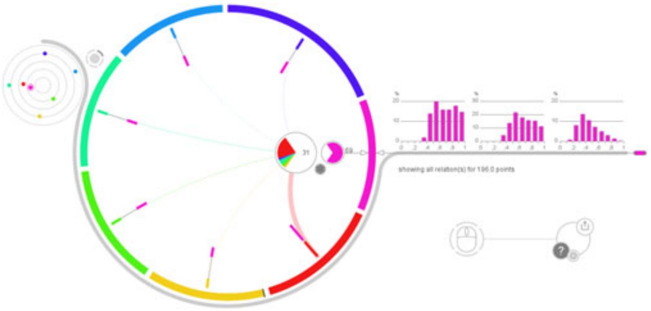

Fig. 5. AUGUR c&r view with activated distance indicators

the merge case, they have equally strong relations to the remaining clusters and are well separated in terms of the radar and distance indicators. Unfortunately, these characteristics are no clear indication for a split, e.g., non-spherical clusters can exhibit the same properties. To gain more certainty in decisions for split candidates, the attribute view has been developed.

Attribute View. When we look at attributes in terms of clustering, we can state the following: If an attribute has a uniform or unimodal distribution (in the following Φ), it is not useful for clustering because the objects of the dataset cannot be clearly separated in this dimension. In contrast, bi- or multi-modal distributions are desired, since they can be used for object separation. When we look at attributes on the cluster level, this is inverted. Regarding a cluster, it is desirable that all of its attributes have unimodal distributions, since this shows high intra-cluster homogeneity. A multimodal-distributed attribute would imply that the cluster could be further separated in this dimension. Generally, we desire the following: On the dataset level, attributes should be dissimilar to Φ, while on the cluster level, they should resemble it as closely as possible. These are the basics for our attribute view.

To calculate the similarity to Φ, we use a straightforward approach. We generate histograms, on the dataset and cluster level, for each attribute. From the histogram bins, those that are local maxima are selected. From each maximum, we iterate over the neighboring bins. If a neigboring bin contains a smaller or equal number of objects, it is counted and the next bin is examined; otherwise, the examination stops. With this, we can determine the maximum number of objects and bins of this attribute that can be fitted under Φ. This is the value we display in the attribute view. In Fig. 6, the attribute view is depicted for the violet cluster E from our example. There are two hemispheres and a band of numbers between them. The band shows the attributes of the dataset, ordered by our computed values, and is used to select an attribute for examination (selection has a darker color). The small hemisphere on the right shows the global behavior of attributes. Each curve represents an attribute, while for the selected attribute, the area under its curve is colored. The hemisphere itself consists of two 90-degree scales, the upper for the percentage of objects and the lower for

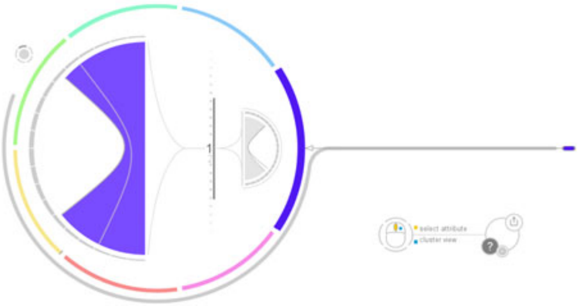

Fig. 6. AUGUR attribute view indicating a split for the violet cluster

the percentage of bins that can be fitted under Φ. The start and end point of each curve show the values for the attribute on these scales. If all objects and bins fit under Φ, a vertical line is drawn and there is no color in the hemisphere. All this also applies to the left hemisphere showing the attribute in the selected cluster. For our example in Fig. 6, we selected attribute 1.

We can see a large colored area, showing that more than 50% of the objects and bins do not fit under Φ. If, in addition, the selected cluster shows split characteristics in the c&r view, the user may assume that this cluster should be split. The benefit of this view lies in the fast and easy interpretability. More color in the left hemisphere indicates a higher split possibility, while the amount of color in the right hemisphere acts as a measure of confidence for the left. In terms of Shneiderman's mantra, this view can either be considered as *'details-on-demand'* or as an *'overview'* and *'zoom and filter'* for the attribute space.

4 Demo Details

The demo at SSDBM comprises an in-depth explanation of all necessary concepts and the demontration of the AUGUR prototype, in which we will show how our visualization and interaction concepts, in tight combination with our flexible aggregation approach, can be used to conduct a visually-driven exploration of scientific data sets. To distinguish our work, we will try to apply some basic and well-known visualization techniques like, among others, scatterplots and parallel coordinates [10] in our described scenario and show their limitations.

We will demonstrate how non-clustering experts are able to efficiently utilize our proposed concepts to determine clustering results with high quality. For this purpose, we will prepare a set of different artificial and realistic data sets (biological domain) to show the applicability of our approach. The artificial data sets will have different degrees of complexity regarding cluster shapes, density of clusters, and outliers. With these data sets, we will simulate different tough situations for data clustering. For the demonstration of the realistic data sets, we will have results being determined by the corresponding data set owner. Using these results, we will show how we can derive the results in a non-expert-oriented way with our AUGUR prototype.

5 Conclusion and Future Work

In this paper, we introduced our AUGUR visualization, which focuses on enabling the user to evaluate an ensemble clustering result and on providing decision support for result refinement with our extended aggregation algorithm proposed in [7]. There already exist a multitude of cluster visualization techniques [10], which mostly try to visualize all objects of the dataset and are thus limited if data sets exceed a certain size. Furthermore, some of these techniques use complex visual concepts, which can hinder interpretation. In contrast, our visualization is tightly coupled to our aggregation method [7]. We do not try to visualize all objects of the data set but concentrate on the presentation of clusters as well as cluster-cluster and cluster-object relations, derived from soft cluster assignments. This result- and relation-oriented approach allows the interpretation of data sets with arbitrary volume/dimensionality and supports the user in making decisions concerning result refinement via the mentioned *split* and *merge* actions. In addition, focusing on *'what'* to visualize, namely clusters and relations, allows the use of well-known and simple visual elements, e.g., pie charts and histograms, when it comes to *'how'* to visualize.

Future work for our AUGUR approach includes the development and integration of a recommender system, additional views on the *details-on-demand* level, and scalability. At the moment, AUGUR can display up to 360 clusters in WXGA resolution.

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. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. ACM Comput. Surv. 31(3) (1999)
3. Jain, A., Law, M.: Data clustering: A users dilemma. In: Pal, S.K., Bandyopadhyay, S., Biswas, S. (eds.) PReMI 2005. LNCS, vol. 3776, pp. 1–10. Springer, Heidelberg (2005)
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. Strehl, A., Ghosh, J.: Cluster ensembles — a knowledge reuse framework for combining multiple partitions. Journal of Machine Learning Research 3 (2002)
7. Hahmann, M., Volk, P., Rosenthal, F., Habich, D., Lehner, W.: How to control clustering results? flexible clustering aggregation. In: Adams, N.M., Robardet, C., Siebes, A., Boulicaut, J.-F. (eds.) IDA 2009. LNCS, vol. 5772, pp. 59–70. Springer, Heidelberg (2009)
8. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum, New York (1981)
9. Shneiderman, B.: The eyes have it: A task by data type taxonomy for information visualizations. In: VL 1996: Proceedings of the 1996 IEEE Symposium on Visual Languages, Washington, DC, USA, p. 336. IEEE Computer Society, Los Alamitos (1996)
10. Hinneburg, A.: Visualizing clustering results. In: Encyclopedia of Database Systems, pp. 3417–3425 (2009)