One Clustering Process Fits All - A Visually Guided Ensemble Approach

Martin Hahmann, Dirk Habich, Maik Thiele, Wolfgang Lehner
Dresden University of Technology
Database Technology Group
01187, Dresden, Germany
dbinfo@mail.inf.tu-dresden.de

Abstract

Looking back on the past decade of research on clustering algorithms, we witness two major and apparent trends: 1) The already vast amount of existing clustering algorithms, is continuously broadened and 2) clustering algorithms in general, are becoming more and more adapted to specific application domains with very particular assumptions. As a result, algorithms have grown complicated and/or very scenario-dependent, which made clustering a hardly accessible domain for non-expert users. This is an especially critical development, since, due to increasing data gathering, the need for analysis techniques like clustering emerges in many application domains. In this paper, we oppose the current focus on specialization, by proposing our vision of a usable, guided and universally applicable clustering process. In detail, we are going to describe our already conducted work and present our future research directions.

1 Introduction

To obtain an optimal clustering, knowledge in the domain of clustering algorithms and the domain of the application data is essential. In practise, the average application users, e.g, biologists, are usually experts of the data domain but only have limited knowledge about the available tools for clustering. Therefore, clustering is a challenging task for this domain experts [Jain and Law, 2005]. The reasons can be briefly summarized as follows: (i) The selection of a clustering algorithm is critical, since, in general, only a fraction of the available algorithms is known to the user. Moreover, most algorithms are tailored to specific tasks and are thus, not appropriate for every data set. (ii) Another obstacle is parameterization, which offers many degrees of freedom but provides nearly no support. (iii) Finally, the interpretation of results, is complicated by the multitude of existing visualization and validation techniques, which are also not universally applicable.

In our opinion, the nearly unlimited options and the high degree specialization of algorithms are the major obstacles, which prevent non-expert users from the successful application of clustering algorithms. Therefore, we state that users first of all need a universally applicable process, rather than a zoo of highly customized clustering algorithm. Based on the paradigm of ensemble-clustering, we want to abandon specialization and develop a unifying clustering process. This process integrates the user, offers guidance and allows the purposeful navigation through the available clustering solutions, as well as the step-by-step construction of a satisfying clustering result, by adjustment of the ensemble clustering and on-demand generation of additional clusterings for parts of the dataset. To realize such an unifed clustering process, research effort has to be done on three areas: algorithms (Section 2), usability (Section 3) and architecture. In the remainder of this paper we will present our research results we obtained so far [Hahmann *et al.*, 2009; 2010b; 2010a] and state the open challenges in the respective research areas (Section 4). We omit the architecture area and refer to the following papers [Habich *et al.*, 2007a; 2007b; 2010]. Finally, we conclude the paper with a brief summary in Section 5

2 Algorithm Area - The Clustering Process

In this section, we introduce the underlying algorithmic plaform for our unifed clustering process—*Flexible Clustering Aggregation (FCA)* [Hahmann *et al.*, 2009]. The basic concept of FCA is clustering aggregation, which combines different clusterings of a dataset into one result to increase quality and robustness [Hahmann *et al.*, 2009; Gionis *et al.*, 2007]. Different aggregation approaches are known, where the pairwise assignment approach is considered as the most capable one [Boulis and Ostendorf, 2004; Caruana *et al.*, 2006; Topchy *et al.*, 2004; Zeng *et al.*, 2002; Dimitriadou *et al.*, 2001; Dudoit and Fridlyand, 2003; Fred, 2001; Fred and Jain, 2003; Frossyniotis *et al.*, 2002; Gionis *et al.*, 2007; Habich *et al.*, 2006]. 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)* [Hahmann *et al.*, 2009] tackles this issue. The key approach of our technique is to change the aggregation input from hard to soft clusterings [Bezdek, 1981]. 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* [Bezdek, 1981] or (ii) calculated from arbitrary clustering results, using refinement techniques like *a-posteriori* [Zeng *et al.*, 2002]. Up to now, we only utilize *fuzzy c-means* to generate the clusterings for our ensembles.

In a soft clustering result, 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 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 ${v_i}^{\top} = (0.4, 0.4, 0.2)$. 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* [Hahmann *et al.*, 2009].

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 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[Hahmann *et al.*, 2009].

Generally, 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 [Hahmann *et al.*, 2009], these *effects* could only be applied to the whole clustering and were thus mutually exclusive. In our recent work, we enhanced the algorithm so that those *effects* can be applied to individual clusters.

3 Usablity Area - Visual Decision Support

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

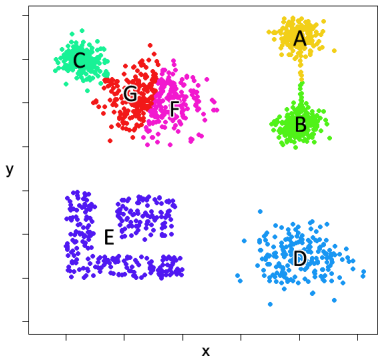

Figure 1: Example aggregate.

To efficiently support result interpretation and adjustments, we developed a visualization concept that is tightly coupled to our aggregation method [Hahmann *et al.*, 2010b]. In general, two major groups of data/clustering visualizations can be distinguished: The first one is data-driven and tries to depict all objects and dimensions of a dataset, which leads to incomprehensible presentations for datasets exceeding a certain scale. The second one is result-driven and thus relatively scale-invariant. For example, a clustering can be depicted as a bar chart showing relative cluster sizes and/or additional values like mean or standard deviation. While the first group often shows too much information, the second one often shows not enough. So, we positioned our approach as a hybrid between those groups, by visualizing the result and the relations between data and result, which are already incorporated in the soft input of our aggregation. In compliance with Shneiderman's mantra, 'overview first, zoom and filter, then details-on-demand' [Shneiderman, 1996], our visualization features three interactive views: overview, cluster composition and relations (c&r), 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 our aggregation algorithm. We define stable clusters according to the general objective of clustering, that asks for clusters with high internal similarity that are well separated from each other. Clusters that not fullfill these criteria are candidates for adjustment. To illustrate our approach, the clustering aggregate depicted in Figure 1 is used as our paper example. It has been generated using our ensemble clustering and shows a partitioning result that needs some adjustments. In all subsequent figures, clusters are identified via color.

3.1 Overview

The overview is the first view presented to the user and depicted in Figure 2. 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 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 Eu-

clidean 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.

3.2 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 Figure 1, 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 Figure 3 (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 [Hahmann *et al.*, 2009]. 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 Figure 3. 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 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 Figure 3 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 Figure 4. 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 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.

3.3 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 Figure 5, 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

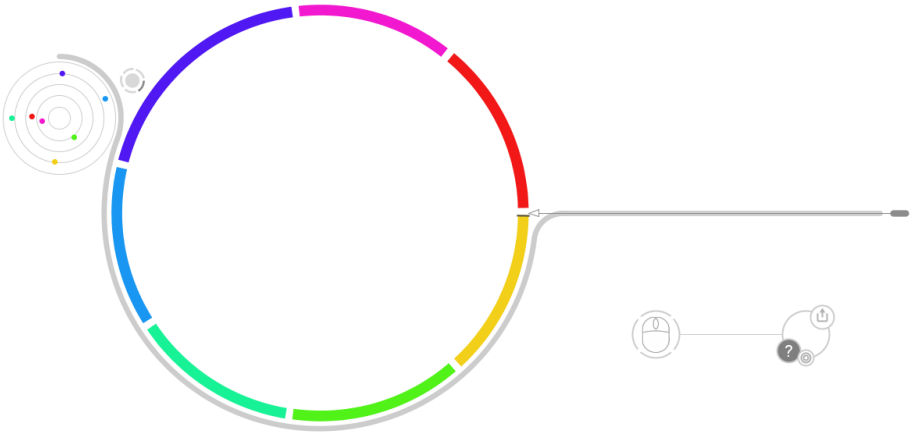

Figure 2: AUGUR overview showing clusters and inter-cluster distances.

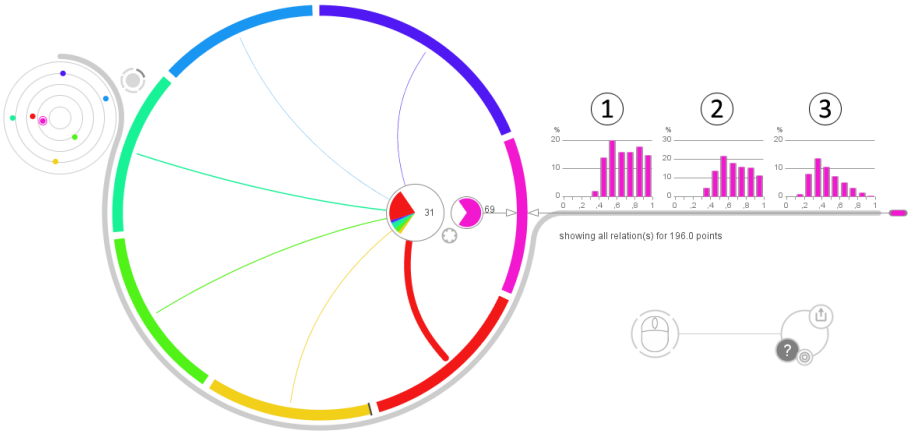

Figure 3: AUGUR c&r view showing composition and relations for the pink cluster.

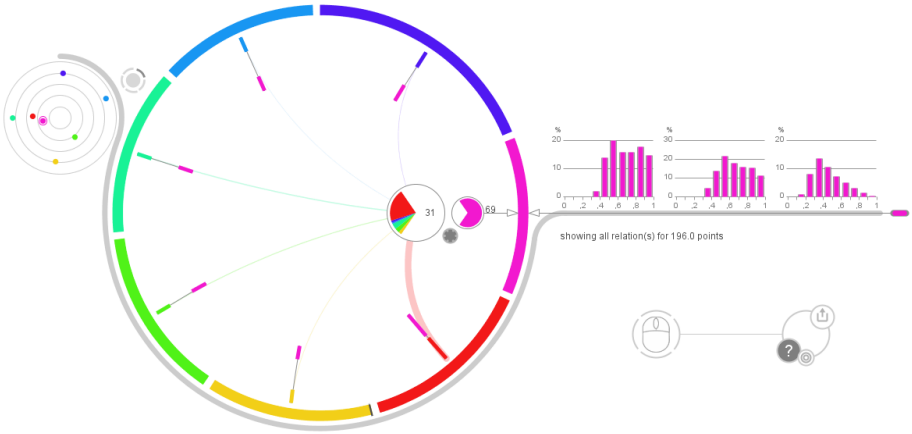

Figure 4: AUGUR c&r view with activated distance indicators.

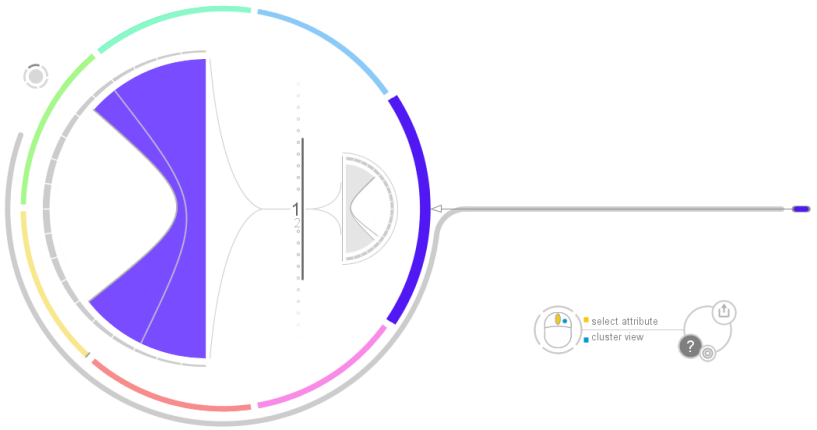

Figure 5: AUGUR attribute view indicating a split for the violet cluster.

curve is colored. The hemisphere itself consists of two 90-degree scales, the upper for the percentage of objects and the lower for 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 Figure 5, 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.

3.4 Feedback

The basic idea of our unifed clustering process is to take advantage of the tight coupling between our two components and change the focus for parameterization from the whole clustering to individual clusters. For this, we use the following workfow: At first, the visualization of an initial clustering aggregate like our running example is presented to the user. In this view, he/she evaluates all clusters and looks for those that need adjustment. For those, the effects merge or split would be identified as appropriate. The respective parameter is then passed to the aggregation algorithm and only the specified clusters are subjected to a new aggregation cycle, while the rest of the result is kept. In sucessive steps, the user adjusts the clustering using the provided feedback operations until the result is satisfying.

4 Future Work

Although, we already acquired several results and components for our unifying process, much work still needs to be done. In the algorithmic area the utilization of the additional information, stored in soft clusterings, has proven beneficial. Therefore we want to expand its employment in our unifying process. Our short term goal is the development of a soft density-based clustering algorithm and a soft hierarchical method later on. In combination with the soft partitioning algorithm we used so far (fuzzy c-means[Bezdek, 1981]), these three algorithms, will provide us with a good coverage, for the generation of our cluster ensemble. To keep parameterization easy, the aggregations role as an abstraction layer must be developed further, so that users are provided with a stable and algorithm-independent interface [Hahmann *et al.*, 2010b] for the adjustment of clusterings. For this it is necessary to specifically implement parameters like *merge* for each algorithm.

Besides parameterization, the whole area of usability will be developed further. The employment of density-based and hierarchical clustering algorithms, leads to different views of the data. Depending on the generating algorithm, different information can be derived from the obtained clusterings. This information must be examined, refined and communicated to the user in the form of novel visualization concepts. Thereby, the focus lies on convenient presentation metaphors, that present information, necessary for the user to navigate through our process without flooding him/her with too much input. In addition, to improve the guidance during navigation, we will implement a semi-automatic recommender system for the selection of appropriate parameters/feedback in arbitrary stages of our unifying process. Another major aspect of our future work in the area of usability will be the integration of novel and intuitive interaction platforms like, e.g., Apple's iPad or Nintendo's Nunchuk controllers. These platforms have already shown in practice, that intuitive interaction simplifies access to unfamiliar technologies.

Concerning the architecture domain, we further focus on scalability for all our used components. Regarding the interaction platform previously mentioned, we will also regard the question, if the components of our process can be distributed and where each component should be executed. An example setting could execute the complete algorithmic stack on high performance hardware, while process interaction is done on a portable device.

In summary, our long term goal, is to combine our existing components with the future work, outlined in this section to form our unifying clustering process. In its final state, this process will provide an initial clustering as starting point, which the user adjusts and refines in a step-by-step fashion. While the user navigates through the process, he/she is offered guidance for parameter and algorithm selection, as well as hints on which parts of the clustering still need refinement.

5 Conclusion

In our work we oppose the specialisation of clustering and claim, that one clustering process can fit all application scenarios. To realize this claim, we concentrate on controllable ensemble-clustering, guided user interaction and emerging computation architectures. We are fully aware, that our position is bold and controversial. Nevertheless, our recent work supports our overall vision of a unifed visually guided clustering process.

In this paper, we summarized our current status on the algorithmic and useability area. We focus 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 [Hahmann *et al.*, 2009]. There already exist a multitude of cluster visualization techniques [Hinneburg, 2009], 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 [Hahmann *et al.*, 2009]. 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.

References

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

[Boulis and Ostendorf, 2004] Constantinos Boulis and Mari Ostendorf. Combining multiple clustering systems. In *Proc. of PKDD*, 2004.

[Caruana *et al.*, 2006] Rich Caruana, Mohamed Farid Elhawary, Nam Nguyen, and Casey Smith. Meta clustering. In *Proc. of ICDM*, 2006.

[Dimitriadou *et al.*, 2001] Evgenia Dimitriadou, Andreas Weingessel, and Kurt Hornik. Voting-merging: An ensemble method for clustering. In *Proc. of ICANN*, 2001.

[Dudoit and Fridlyand, 2003] Sandrine Dudoit and Jane Fridlyand. Bagging to improve the accuracy of a clustering procedure. *Bioinformatics*, 19(9), Jun 2003.

[Fred and Jain, 2003] Ana L. N. Fred and Anil K. Jain. Robust data clustering. In *Proc. of CVPR*, 2003.

[Fred, 2001] Ana L. N. Fred. Finding consistent clusters in data partitions. In *Proc. of MCS*, 2001.

[Frossyniotis *et al.*, 2002] Dimitrios S. Frossyniotis, Minas Pertselakis, and Andreas Stafylopatis. A multi-clustering fusion algorithm. In *Proc. of SETN*, 2002.

[Gionis *et al.*, 2007] Aristides Gionis, Heikki Mannila, and Panayiotis Tsaparas. Clustering aggregation. *TKDD*, 1(1), 2007.

[Habich *et al.*, 2006] Dirk Habich, Thomas Wächter, Wolfgang Lehner, and Christian Pilarsky. Two-phase clustering strategy for gene expression data sets. In *Proc. of SAC*, 2006.

[Habich *et al.*, 2007a] Dirk Habich, Steffen Preissler, Wolfgang Lehner, Sebastian Richly, Uwe Aßmann, Mike Grasselt, and Albert Maier. Data-grey-boxweb services in data-centric environments. In *Proceedings of the 2007 IEEE International Conference on Web Services (ICWS 2007, July 9-13, Salt Lake City, Utah, USA)*, pages 976–983, 2007.

[Habich *et al.*, 2007b] Dirk Habich, Sebastian Richly, Mike Grasselt, Steffen Preißler, Wolfgang Lehner, and Albert Maier. Bpel^{DT} - data-aware extension of bpel to support data-intensive service applications. In *Proceedings of the 2nd Workshop of Emerging Web Services Technology in conjunction with the 5th IEEE European Conference on Web Services (WEWST 2007, November 26, Halle/Salle, Germany)*, 2007.

[Habich *et al.*, 2010] Dirk Habich, Wolfgang Lehner, Sebastian Richly, and Wolfgang Lehner. Using cloud technologies to optimize data-intensive service applications. In *Proceedings of the 3rd International Conference on Cloud Computing (CLOUD 2010, Miami, FL, USA, June 5-10)*, 2010.

[Hahmann *et al.*, 2009] Martin Hahmann, Peter Volk, Frank Rosenthal, Dirk Habich, and Wolfgang Lehner. How to control clustering results? flexible clustering aggregation. In *Advances in Intelligent Data Analysis VIII*, pages 59–70, 2009.

[Hahmann *et al.*, 2010a] Martin Hahmann, Dirk Habich, and Wolfgang Lehner. V2i: A model process for accessible clustering. In *Appears as Demo at the 16th ACM SIKDD Conference on Knowledge Discovery and Data Mining (KDD 2010, Washington, DC, USA, July 25-28)*, 2010.

[Hahmann *et al.*, 2010b] Martin Hahmann, Dirk Habich, and Wolfgang Lehner. Visual decision support for ensemble clustering. In *Proceedings of the 22nd International Conference in Scientific and Statistical Database Management (SSDBM 2010, Heidelberg, Germany, June 30 - July 2)*, pages 279–287, 2010.

[Hinneburg, 2009] Alexander Hinneburg. Visualizing clustering results. In *Encyclopedia of Database Systems*, pages 3417–3425. 2009.

[Jain and Law, 2005] Anil Jain and Martin Law. Data clustering: A users dilemma. *Pattern Recognition and Machine Intelligence*, pages 1–10, 2005.

[Shneiderman, 1996] Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *VL '96: Proceedings of the 1996 IEEE Symposium on Visual Languages*, page 336, Washington, DC, USA, 1996. IEEE Computer Society.

[Topchy *et al.*, 2004] Alexander P. Topchy, Behrouz Minaei-Bidgoli, Anil K. Jain, and William F. Punch. Adaptive clustering ensembles. In *Proc. of ICPR*, 2004.

[Zeng *et al.*, 2002] Yujing Zeng, Jianshan Tang, Javier Garcia-Frias, and Guang R. Gao. An adaptive meta-clustering approach: Combining the information from different clustering results. In *Proc. of CSB*, 2002.