Interactive Clustering: A Comprehensive Review

JUHEE BAE and TOVE HELLDIN, University of Skövde, Sweden
MARIA RIVEIRO, Jönköping University, Sweden and University of Skövde, Sweden
SŁAWOMIR NOWACZYK and MOHAMED-RAFIK BOUGUELIA, University of Halmstad, Sweden
GÖRAN FALKMAN, University of Skövde, Sweden

In this survey, 105 papers related to interactive clustering were reviewed according to seven perspectives: (1) on what level is the interaction happening, (2) which interactive operations are involved, (3) how user feedback is incorporated, (4) how interactive clustering is evaluated, (5) which data and (6) which clustering methods have been used, and (7) what outlined challenges there are. This article serves as a comprehensive overview of the field and outlines the state of the art within the area as well as identifies challenges and future research needs.

CCS Concepts: • **Computing methodologies → Cluster analysis**; • **General and reference → Surveys and overviews**;

Additional Key Words and Phrases: Clustering, interactive, interaction, user, evaluation, feedback

ACM Reference format:
Juhee Bae, Tove Helldin, Maria Riveiro, Sławomir Nowaczyk, Mohamed-Rafik Bouguelia, and Göran Falkman. 2020. Interactive Clustering: A Comprehensive Review. *ACM Comput. Surv.* 53, 1, Article 1 (February 2020), 39 pages.
https://doi.org/10.1145/3340960

1 INTRODUCTION

While analyzing data, a widely used task is to find groups of dataset objects that share similar characteristics. In doing so, users gain insight into their data, understand it, and even reduce its high-dimensionality nature. These conceptual groups are commonly referred to as clusters. Automatic clustering is a technique that discovers "natural" structures hidden in the data in an unsupervised way. It consists of automatically grouping a set of unlabeled data samples into clusters so that samples in the same cluster are more similar to each other than to samples assigned to other clusters. Although a cluster is inherently a subjective structure, without a

This work has been carried out under grant BIDAF 20140221 (A Big Data Analytics Framework for a Smart Society), funded by the Swedish Knowledge Foundation and EXPLAIN VR 2018-03622 (Evaluation of eXplainable Artificial Intelligence), funded by the Swedish Research Council.
Authors' addresses: J. Bae, T. Helldin, and G. Falkman, University of Skövde, School of Informatics, P.O. Box 408, SE-54128, Skövde, Sweden; emails: {juhee.bae, tove.helldin, goran.falkman}@his.se; M. Riveiro, Jönköping University, Department of Computer Science and Informatics, School of Engineering, Gjuterigatan 5, 55111 Jönköping, Sweden and University of Skövde, School of Informatics, P.O. Box 408, SE-54128, Skövde, Sweden; emails: maria.riveiro@ju.se, maria.riveiro@his.se; S. Nowaczyk and M.-R. Bouguelia, University of Halmstad, School of Information Technology, Kristian IV:s väg 3, 30118, Halmstad, Sweden; emails: {Slawomir.Nowaczyk, mohamed-rafik.bouguelia}@hh.se.

0360-0300/2020/02-ART1
https://doi.org/10.1145/3340960

precise and formal definition, a large number of clustering methods have been developed, each with its particular weaknesses and strengths [109].

Traditionally, clustering methods and tools have been designed offline and then deployed in a variety of application domains. However, because such tools lack domain-specific and user-specific input, they are not always as relevant or convenient to the end-user as they could be. There are several reasons for this. First, unlike classification tasks that are evaluated using well-defined target labels, clustering is, as mentioned earlier, an intrinsically subjective task as it depends on the interpretation, need, and interest of users. Real-world data may contain different plausible groupings, and a fully unsupervised clustering has no way to establish a grouping that suits the user's needs, because this requires external domain knowledge. Second, quality of a clustering outcome is heavily dependent on extracting appropriate features as well as specifying appropriate similarity measures. In addition, several parameters are typically required—for example, the number of intended clusters or the minimum cluster size. Given these requirements, a real-world clustering task can be too complex to be solved fully automatically. Fortunately, a small amount of user input can often significantly help to achieve a better clustering quality. Third, a large part of "understanding data" is to understand the clustering process by which these conceptual clusters are formed. For this purpose, end-users are usually motivated and willing to interact with both the system and the data in a way that let them gain knowledge from the clustering task.

For all these reasons, there is an increasing need for methods that engage end-users directly into the clustering process to tailor it to specific application domains and allow it to continuously adapt to their preferences. Toward this goal, several interactive clustering approaches have recently emerged in which the user and the system interact with each other to carry out the clustering task. Efforts for designing such interactive clustering techniques have been pursued by several research communities including data mining, visual analytics, machine learning, and human-computer interaction. Surprisingly, even though there is a significant number of papers about interactive clustering, the field still lacks a comprehensive survey on different types of interaction and why interactive clustering can be preferred in scenarios where usual (automatic) clustering methods are less helpful. This article provides a comprehensive review of interactive clustering methods and tools to fill this gap.

We propose to untangle the different components of interactive clustering to better understand current state of the art as well as to outline the most important challenges and future research needs. For this purpose, the reasons why interactivity is important in the clustering process are presented in Section 2 based on the motivations provided in different papers from the literature. Interactive clustering approaches are then grouped according to three criteria corresponding to a three-dimensional design space: (i) at which stage is the interaction happening (Section 3), (ii) which interactive operations are involved both from the user and the machine perspectives (Section 4), and (iii) how user's feedback is incorporated to improve the clustering model (Section 5). In addition, Section 6 gives perspectives on subjective and objective evaluation metrics used in the interactive clustering literature. Section 7 raises an issue of the term interactive clustering compared with clustering with interaction. Section 8 classifies papers according to the datasets and the application domains, and Section 9 arranges papers according to the baseline clustering algorithms that have been employed or modified for interactive clustering. Major challenges related to interactive clustering, future directions, and topic modeling result on collected abstracts are discussed in Section 10.

Survey methodology and categorization process

To collect relevant papers, the initial search was performed by the authors with the search words "interactivity," "interaction," and "clustering." Building upon this initial list, we used *Publish or*

Perish [51] software to search papers indexed with Google Scholar based on the following combinations of words: "interactive" and "cluster"; "interactive" and "topic" and "modeling"; "interactivity" and "clustering"; "visual" and "analytics" and "clustering"; "visual" and "analytics" and "cluster." To focus the survey on recent work, we limited the results to papers published in year 2000 or later except for one paper from 1997 [95]. We have also searched for "active clustering" and "active cluster" but found them generally irrelevant for this survey. In fact, they all focused on the use of clustering as a step in active learning process and were therefore not relevant for this survey. Overall, we analysed 1,430 papers, of which we have ultimately decided to include 105.

We have used the following criteria to select the papers. Primarily, the paper's main contribution had to be related to the development or evaluation of clustering algorithms. We have decided not to include papers where clustering was just one step in a larger solution, nor those that simply used state of the art methods. Additionally, the interactive clustering aspect needed to be highlighted as one of the goals. Finally, from series of papers that were clearly building upon each other, we have generally included the latest one unless a particular earlier version provided a clear contribution to this survey.

The selection process was carried out iteratively. First, based on the article's title, abstract, keywords, publication venue, and citations per year, the 1,430 papers were classified under three categories: relevant, not relevant, and maybe relevant. The 1,209 papers marked as not relevant were discarded. In the second round, the "maybe relevant" articles were discussed between the authors, and based on an assessment of the full-text article, they were placed under the other two categories. The 105 papers reviewed were analyzed with respect to several perspectives referred to as *dimensions* described in the title of each section. The dimensions were outlined after the first round of review and refined during the whole process of review and analysis.

1.1 Terminology and Definitions

There are various, often conflicting, definitions of interactive clustering in the literature. For instance, interactive clustering is defined by Chuang and Hsu ([32], p. 2) as "the user-driven process of refining and optimizing the clusters for subsequent analyses." In general, interactive clustering is used when the user holds an active position or role in the clustering process [16]. However, many papers use the term without defining it.

The terms "interactive," "interactiveness," and "interaction" appear in multiple contexts and are used for different purposes. Some papers, such as Zhou et al. [115], used "interactive" or "interactive system" when a user can "explore" the results and a system allows the user to visualize, confirm, modify, accept, or reject. Accordingly, "interactive" is used when a user can preset the value of parameters (e.g., number of clusters). As Chuang and Hsu [32] emphasize the user-driven support (human-in-the-loop), they also argue that explainability is crucial in "interactive clustering," since users usually want to understand how the input features lead to the final result. Many times, it is considered "truly" interactive when the user is satisfied with the clustering result and the clustering quality (i.e., meaningfulness, reliability, interpretability) is good enough based on the objective/subjective criteria used. Others refer to "interactive clustering algorithms," assuming that there is a user who already has the target clustering that aligns to "a user can confirm" statement mentioned above. Balcan and Blum [13] and Awasthi and Zadeh [10] propose a system that learns from user's feedback, which reaches to true clustering with as few queries as possible. The algorithm or system proposes a clustering of the data and requests user's feedback (the correctness of the current hypothesis), which can continue multiple times until the user is satisfied with the clusterings suggested by the system.

The intent of this survey was to focus on papers where the user is actively involved in the clustering process, i.e., those that go beyond interactive visualization of the results. However, as we have began studying the literature, we found out that the boundary was not as crisp as expected. In particular, it was often impossible to make the judgment based on the abstract of the article. Therefore, we believe it is important that a survey like this points out this issue, since there is a definite need to increase the clarity and consistency of presentations in the field.

2 WHY INTERACTIVE CLUSTERING IS HELPFUL

There are several different reasons for favoring interactive clustering that can be broadly divided objectively and subjectively. It corresponds to some degree to the evaluation measures in Section 6, since the motivation should be reflected in the evaluation measure used. However, the resemblance is not absolute, therefore we believe it is worthwhile to delve deeper into the motivations for involving users instead of fully automating the clustering process.

In particular, we identified four categories in this section. The first and most concrete goal is to achieve better clustering quality under the assumption that the task is too complex to be completely automated. Second, to better understand the data and the result, involving the user early in the process, which leads to better understanding of the intricacies of the solution when compared with simply presenting the result. The third goal is to identify interesting data; the ultimate goal here is not the clustering result itself but rather what it says about the data—finding outliers, regions of interest, and so on. Fourth, the reason for the interaction may be related to the subjectivity of the clustering task—that there is no single objective quality measure but rather that the final solution must match the user's needs and expectations.

In the end, of course, all these motivations correspond to "better clustering" in some sense. However, intentional differences often translate into quite different approaches from the technical and algorithmic perspectives. It is impossible to compare or understand the rationale behind different methods without understanding this aspect.

2.1 Improving the Clustering Quality

The most popular goal is to maximize the quality of clustering results; according to several papers, for example, in Chuang and Hsu [32], this is the very purpose of interactive clustering. The basic assumption made by the authors of the papers in this category is that an interactive and collaborative process combining the strengths of both human and machine would yield better results than a process that is purely automated or purely manual. Several examples of improving the quality of the clustering results using different strategies are given in the works presented in Andrienko and Andrienko [4], Basu et al. [15], Boudjeloud-Assala et al. [19], Cao et al. [24], Castellanos-Garzón et al. [26], Choo et al. [30], Dobrynin et al. [38], Hadlak et al. [50], Hoque and Carenini [53], Hu et al. [55], Kumpf et al. [64], Lai et al. [66], Lee et al. [67], Lei et al. [68], MacInnes et al. [72], Packer et al. [79], Schreck et al. [86], Srivastava et al. [94], Turkay et al. [99, 101], Zhou et al. [116].

An example is Siirtola [92], where two user interfaces are compared, enabling the user to cluster datasets and subsets of variables or instances. The user can interactively refine the parameters with instructions to solve the tasks "as quickly as possible while trying not to make mistakes." Similarly, Looney [70] shows that allowing a user to interactively adjust a parameter for merging clusters can improve the quality of the clustering result. On the contrary, in Wang and Davidson [107], spectral clustering is extended to include user interactions in the form of queries that significantly outperforms the baseline approach. Wu et al. [110] experimentally shows that by taking advantage of user interactions, the recall improves consistently across all the domains they have tested with close to 15% gain in the best case and 2.9% on average; the average increase in precision is 7.8% and in F-measure 5.3%. Chen and Liu [28] claims that involving users leads to higher-quality results, especially in the case of clusters of arbitrarily complex shapes; however, the paper provides no formalization of this concept, but experimental results in Chen and Liu [29] show improvements in both the quality of the intermediate clustering results and in extending the intermediate results to a large dataset. In Andrienko and Andrienko [4], the users refine the clusters by selecting distance functions from a library that describe density-based algorithm, OPTICS, according to the goals of the analysis. Schreck et al. [86] shows how analysts refine the quality of clusterings based on expert knowledge. The analysts define new patterns for each node of the SOM grid using SOM algorithm and build user-defined trajectory cluster maps. As a means to improve the quality of the clusters in MacInnes et al. [72], the analysts can provide suggestions for centroids of k-means clusters. User refinement is also prevalent in El-Assady et al. [42], where a tool is presented for users to refine the topics generated by, for example, boosting the impact of certain keywords and by using a better-performing model as an "anchor" for the next learning iteration. This tool also guides the user toward the most ambiguous documents where user feedback would have the most impact on topic modeling.

In general, the expected benefits of interaction are often, implicitly or explicitly, based on the assumption that the users can supply domain knowledge to provide insights that are not captured by the data itself [67, 73, 75, 81, 85], for example in biomedical [47], picture [76], or text data [62, 77]. A clear example is Wu et al. [110], where clustering is used to find matches among fields in two different deep-web query interfaces—in such a setting, the user improves the results beyond what can be done in a purely data-driven way. In Kwon et al. [65], the user's expertise guides the clustering algorithm toward the right size and type of clusters. Bruneau et al. [23] allows the user to "easily inject his or her domain knowledge progressively." The authors in Huang et al. [59] use a visual representation of the clustering tree for refinement and validation purposes, because, based on their knowledge, the users accept or reject partitions. Users in Berthold et al. [17] and Okabe and Yamada [78] insert expert knowledge for cluster refinement and fine-tuning, e.g., through interactive constrained clustering.

A common argument is that a human (or user) must be kept in the loop in the clustering process, since clustering is difficult. That is, it requires knowledge of the analysis domain as well as task-specific adaptations, making it a process that cannot be fully automated [32]. A narrower view is presented in Turkay et al. [99], which focuses on clusters at various stratification levels–accurate clustering at various levels is hard to achieve without interactive analyses.

Related to the quality of clustering results is the quality of the clustering process. We highlight Ailon et al. [2], which uses a query mechanism to reduce computational complexity, proposing a polynomial solution to an NP-hard problem. In Xiao and Dunham [111], the authors emphasize the need for an interactive clustering process for transactional data based on the fact that in many cases such data "may not be provided all at once to the clustering algorithm." One example could be web search engine's results, where new data is delivered through the network in a continuous fashion and at varying speeds. A tool that dynamically presents current clusters as soon as they are produced while the algorithm is running helps to "give the user a rough idea about what the final clusters will be"; that way, the user is able to continuously adjust clustering parameters, for example, the number of clusters that is unknown *a priori*.

The refinement operations are accommodated through various means. In Guo et al. [49], users incorporate their knowledge of the data by identifying potential clusters and refining them through the attractive/repulsive operators and by joining and deleting clusters in a graph view. Similarly, Erra et al. [43] and Awasthi et al. [9] allow splitting and merging the clusters; in particlular, Erra et al. [43] allows the users to interact with the data points that can yield immediate clustering results using GPU. In both Choo et al. [31] and Sourina and Liu [93], the users are enabled to compare and specify various clustering parameters, enabling them to find the best clusters based on their domain knowledge. Various visualizations are also used to let the users refine the clustering results. For example, in Gaudin and Quigley [45], a user is presented with a node-link graph and matrix view of the clusters. If the node-link diagram is too cluttered or too small, then the user can adapt the clustering parameters accordingly, changing the settings for how likely nodes are to end. The refinement operations are enabled through histograms [36], scatterplots [21], and node-link graphs that display to what degree the clustering algorithm was able to accommodate the constraints defined by a user. The users may interact with the algorithm to select the size of the subsets [108] or automatic parameter estimations [61] to incrementally refine the clustering result.

2.2 Understanding the Clustering Process

The broadest reason for interactivity is to make it easier for users to understand the final result, see, for instance, Basu et al. [15], Boudjeloud-Assala et al. [19], Cao et al. [25], Choo et al. [30], Hoque and Carenini [53], Lee et al. [67]. One clear example is autonomous data exploration assistant (AIDE) [95], which can perform clustering as it incrementally explores a dataset. The users do not interact with the clustering process directly, but they can ask AIDE to provide explanations regarding its choice of data and parameters. This is common in papers that focus on data exploration such as Brandt et al. [20] and other examples already described in earlier subsections under this dimension, e.g., Andrienko and Andrienko [4, 5], Cao et al. [24], Dobrynin et al. [38], Hadlak et al. [50], Hossain et al. [54], Hund et al. [60], Lei et al. [68], L'Yi et al. [71], MacInnes et al. [72], Packer et al. [79], Sarnovsky [84], Turkay et al. [101], Zhou et al. [116]. Several solutions in this category highlight the importance of being highly reactive, e.g., in the ICEAGE system described in Guo et al. [48]. The results in several plots are instantly updated using different colors for different clusters after the user explores the hierarchical clusters. Another way is with innovative iconic-visualizations of the clusters, like those shown in Cao et al. [24].

In other cases, such as El-Assady et al. [42], the focus is to enable the user to understand and adapt the clusters (e.g., topic models) used without having a deep understanding of the algorithms or having to read all the text corpora. Several authors Choo et al. [31], Chuang and Hsu [32], Guo et al. [49], Sourina and Liu [93] argue that user interaction is the key to understanding and interpreting complex datasets and patterns, i.e., what a cluster means and how one cluster differs from another. A specific example from the medical domain is Kwon et al. [65], where the three highest quality clusters are used to understand why certain data points (i.e., patients) are clustered together or separately. For temporal data, Turkay et al. [100] proposes a "temporal cluster view" and "temporal signatures" for that purpose.

A common technique to increase understanding of the results is to present multiple alternative results and make it as easy as possible for the user to choose the "correct" or "preferred" one [62, 75, 89]. For example, in Siirtola [92], the user explores both hierarchical and partitional clusterings of a dataset, with an indication of the overlap. Xclusim [71] allows users to understand the clusters generated by comparing different and alternative solutions generated by many clustering techniques embedded in the tool.

An alternative is to present a single clustering result but to use visualization techniques that simultaneously present many aspects of the result [37, 88, 113]. For example, Van Long and Linsen [102] focuses on comprehension of the clusters' structure from the multi-dimensional multivariate data using parallel coordinates.

2.3 Find Interesting Data

A goal that is related yet subtly different to "understanding" is finding interesting data. In a sense, these papers consider the whole clustering process as a means to an end—clustering activity or clusters themselves are not an ultimate goal. Rather, they are useful for highlighting data of particular importance (e.g., Andrienko et al. [6], Boudjeloud-Assala et al. [19], Bruneau and Otjacques

[22], Jiang and Canny [62], Lee et al. [67], Rawlins et al. [83], Seo and Shneiderman [88], Turkay et al. [99]).

Two cases using movement data utilize clustering to help users discover interesting locations based on travel trajectories (Zhou et al. [115], personal gazetteer discovery problem) and to identify anomalous trajectories (Andrienko et al. [4, 6]). In the medical domain, Kwon et al. [65] provides insights to the data regarding the patients who have been diagnosed with the same disease but respond to the treatments differently, which may encourage users to provoke new questions to the clustering results. In Andrienko et al. [7], users interactively find interesting traffic patterns (e.g., typical commuter routes in a city) with spatial and spatio-temporal data.

Another example is a system for ecologists, presented in Ahmed and Weaver [1], focusing on large number of data measurements in "development and validation of complex ecological models" ([1], p. 1). In this case the most interesting property is the *variation at different time scales*. Clustering is a very useful tool, but the ultimate goal in this context is not to find good clusters but to interactively analyze the ecological data. The activity includes selection of variables of interest for clustering and display, to choose and fine-tune the timescale (e.g., days, years, seasons), or to shift the designated time frame back and forth to check the updates.

Yet another example that uses interactive clustering for knowledge discovery is Wan et al. [105], where interaction is used to support the hypothesis of existence (or non-existence) of transcription and translation patterns in genomes. This is commonly done as part of data exploration, i.e., allowing the user to better understand the data and identify interesting subsets of the data [20, 102].

2.4 Subjective Clustering

The final reason to involve users in a clustering process is the inherent subjectivity of the task. The overall idea is that there are many different possible clusterings for the same data, and there is no clear best one. However, the users have certain expectations (or needs), and the goal of the interactive process is to support them in finding the clustering that is the most suitable for their individual goals, see, for instance, Andrienko et al. [6, 7], Basu et al. [15], Cao et al. [24], Chang et al. [27], Dobrynin et al. [38], Dubey et al. [40], Iorio et al. [61], Lee et al. [67], MacInnes et al. [72], Packer et al. [79], Schreck et al. [86], Xu et al. [113], Zhou et al. [116]. Many of the interactive frameworks presented in the aforementioned references support the user performing subjective clustering based on expert's knowledge. Srivastava et al. ([94], p. 1) notes, "What makes it challenging to identify a good clustering is that it is often difficult to encode the analyst's goals explicitly as machine learning objectives," because "what constitutes a good clustering may depend on domain-specific and application-specific criteria," but "it seems reasonable to expect that the analysts will know a good clustering when they see one." This is common, for example, in search tasks or in document clustering [16, 69, 97]. According to Chang et al. ([27], pp. 3 and 4), search results' clustering is challenging to fully automate due to a number of requirements: The algorithm must produce clusters with "topical coherence," "semantically meaningful topic labels," maximize cluster separation, achieve appropriate granularity of clusters, ensure balanced cluster sizes, and guarantee high coverage of search results. Arguably, human expert judgment is needed to balance these requirements by steering the clustering process and evaluating the results [54]. In the same context, in Bekkerman et al. [16], Choo et al. [30], Corrêa et al. [34], El-Assady et al. [42], Hoque and Carenini [53], Hu et al. [56–58], the interaction supports generating clusters that are based on user's domain knowledge in accordance with their understanding to get clusters that fit user's expectations well. In Awasthi and Zadeh [10], the following scenario is given as motivation: "Consider documents representing news articles that could be clustered as {politics, sports, entertainment, other}; however, perhaps the user would like these articles to be clustered into {news articles, opinion pieces}" ([10], p. 1).

There have been several attempts to formalize this concept. Mukhopadhyay et al. [74] proposes an interactive multi-objective clustering algorithm based on a weighted set of measures on cluster validity. A number of high-quality clustering solutions are periodically presented to a user, and the non-conforming validity measures get penalized by users' ranking based on preferences. This is somewhat in contrast to Xiong et al. [112], which assumes that user query is a way to retrieve one "correct" clustering solution given imperfect and noisy data and distance/similarity metrics.

However, in most cases there is no formal problem definition, and the task is expressed only in an intuitive way. In Vikram and Dasgupta ([103], p. 1), their motivation comes from addressing the difficulty of clustering and to "find a grouping that aligns with a user's needs." Given that in many complex datasets there are several plausible clusterings, different users may have different priorities and preferences and an unsupervised algorithm has no way to intrinsically guess these. In Arın et al. [8], the user merges clusters until satisfied. In Coden et al. [33], the tasks, such as grouping food items into categories for a restaurant menu, are presented as subjective and lacking both the feature set and well-defined dis-/similarity metrics. In most cases [11, 20, 31, 35], the user iteratively re-defines the clustering tasks (e.g., focusing on a subset of data points and attributes) and refines the clustering parameters (e.g., algorithm, number of clusters, threshold), until a satisfactory solution is reached.

3 AT WHICH STAGE IS THE INTERACTION HAPPENING

Hinneburg [52] observes that when it comes to combining automated clustering algorithms with interactive visualization techniques, the general design pattern is to replace some part (or stage) of the algorithm with an interactive visual procedure that allows a user to affect the final outcome. There are different ways in which such combinations could be organized, and in this article we decided to divide it into three main groups. The following structure was inspired by Hinneburg [52]; however, not all papers in our survey fitted the taxonomy suggested there, which we find differences.

The first group is in some sense the most advanced and closest to Hinneburg's "automated clustering." The user interacts with the clustering results by providing feedback concerning mistakes and hints at an improved solution. The second group is similar, except that user's interactions are explicitly directed at model or algorithm parameters rather than at clustering results. In other words, the former (i.e., interact with results) needs to "translate" user's intentions into concrete actions applicable to the clustering algorithm, while in the latter (i.e., interact with model), those intentions are directly expressed in terms of algorithm parameters (somewhat in line with Hinneburg's "model selector"). In both of these groups, the initiative rests firmly with the user, i.e., it is the user who decides on the feedback. An alternative is our third group, where the clustering algorithm explicitly queries a user by asking for appropriate input (such a category has not been discussed by Hinneburg).

The concept of different interaction stages is not limited to clustering, of course. Our structure of this dimension matches quite well with the types of interaction identified by Self et al. [87]. They distinguish "Parametric interaction" and "Observation-level interaction," with the former referring to users directly specifying or modifying design parameters of an algorithm, and the latter allowing users to interact with individual data items, usually in an interactive graphical tool. Self et al. [87] argue that these two forms of interaction offer distinct and complementary capabilities, and are likely to lead to different types of insights. Parametric interaction offers high degree of control but requires deep understanding of the analytical model; while observation-level interaction offers familiar interface embedded in the domain semantics, but changes made by the user may be incorrectly translated into model updates. However, and somewhat surprisingly, in the field of interactive clustering most of the papers focus on one or the other, and such combinations are not common.

The details of these three groups are presented in the following subsections. For this survey, observe that we excluded Hinneburg's "purely manual" category, where there is no clustering algorithm and the clusters are constructed directly by the user through different means of interactive visualization.

3.1 Interacting with the Result

As stated by Chuang and Hsu [32], the functionality to allow the user to iteratively and dynamically refine the clustering results through meaningful operations is of utmost importance to achieve a more user-based clustering process. The most complex type of interaction is for the machine to perform (initial) clustering, present the results to the user through visualization, and only then give a user the option to interact with clustering results. The exact operations that the user is allowed to perform vary (see Section 4), but users are commonly expected to identify and fix any mistakes and imperfections in the clustering results. Such fixes provide to the machine/model/algorithm hints about user's preferences or about properties of the task at hand. This feedback is then analyzed and understood to appropriately propagate the new information and, in effect, arrive at better overall result. For example, if the user identifies a mis-assigned data point, then the assumption is that it is not enough to move this individual point to another cluster—the algorithm should generalize this observation and learn from it by figuring out what a better optimized clustering is that incorporates the new information. This can be contrasted with the second-most-common approach, "interaction with model parameters," in which the middle step is done instead by human/user; it is the user who needs to identify which parameters must be modified to get a desired effect.

Conceptually, the algorithm provides visual information to the users to better understand the clusters. Through users' expertise and domain knowledge, they expect to influence and steer the clustering process toward better results—without necessarily needing to understand the inner workings of the algorithm. The identification of incorrect algorithm parameters (e.g., number of clusters, instance-similarity measure) is done automatically; the clustering algorithm then re-runs with more appropriate values—hopefully leading to a better solution. Such a cycle can be repeated many times, with the basic idea of feedback operations automatically triggering a re-computation of the algorithm.

The one interaction technique that we have actually found common is adding must-link/cannot-link constraints [21, 36, 55]. The semantics of these constraints is quite clear and generally leads to updating the distance matrix. But authors disagree with respect to the "strength" of such constraints, with some assuming them to be hints or guidelines to be weighted against other quality criteria (e.g., desJardins et al. [35], Muller et al. [76]) while others consider them to be unbreakable (e.g., Gruźdź et al. [47], Lai et al. [66]). Either way, it is natural to maintain those constraints across several iterations of interactive clustering. For example, Gruźdź et al. [47] updates the self-organizing maps (SOM) that encode similarities and dis-similarities between pairs of genes corresponding to their expression characteristics. However, only a few solutions (e.g., Okabe and Yamada [78]) provide hints to the user for a better selection of constraints, which reduces the number of constraints required. An extension of this approach, especially popular in the topic modeling domain, allows the user to split or merge clusters without specifying individual data instances for must-/cannot-link constraints [9, 10, 13, 19, 30, 53, 54, 69].

The rest of the papers provide more unique solutions that often—despite similar names and high-level descriptions—can have different semantics, and it is unclear in some papers how the feedback from such operations is maintained over time. One such example is allowing the user to move data instances from one cluster to another (e.g., Basu et al. [15], Coden et al. [33]). Dubey et al. [40] calls this "assignment feedback" and claims that manually moving data points between clusters supports the exploration of high-dimensional data. For relational data, desJardins et al. [35] allows the user to relocate "misplaced" instances. A different approach, called "cluster description feedback," is presented in Dubey et al. [40], where a feature vector corresponding to a single cluster is modified by a user "to make it more meaningful." The approach presented in Mukhopadhyay et al. [74] learns users' preferences for relative weighting of different clustering quality criteria based on users' ranking of several example solutions from the current Pareto frontier. The TINDER approach of Srivastava et al. [94] presents clusters to the user one by one for inspection, and the user is given an opportunity to reject a cluster if it does not meet their expectations. The objective function of the clustering algorithm is then modified to guarantee that subsequent clusters are as dissimilar from the rejected ones as possible, while maintaining high intrinsic quality.

Finally, interacting with clustering results as described in several of the papers consist of different combinations of adapting various parameters in more or less independent ways (refer to Section 3.2). Bruneau et al. [23] uses the distribution of the data to transform the high-dimensional clustering, updates the two-dimensional (2D) embedding by dissimilarity transform, and allows users to limit operations to a subset of data. In Xu et al. [113], users interact with node-link diagrams, adjacency matrices, and tree-maps to refine clusters, either by directly interacting with the visual representation (e.g., remove a node from a cluster or relocate a node to an appropriate cluster) or by updating the similarity measure. In Andrienko et al. [7], the user can review and revise the classifier using the visual representations of sub-clusters to remove cluster members, split clusters into smaller ones, merge clusters, and redistribute cluster members. The AppGrouper tool of Chang et al. [27] allows interaction at multiple stages: Refine input to the clustering algorithm, control the granularity of resulting clusters, and adapt topic labels. Labeling sub-clusters is also supported by Mitchell [73]. The 3D-VisualCluster tool of Castellanos-Garzón et al. [26] allows users to interact with clustering results by finding and selecting boundary points of previously grouped clusters. In Andrienko et al. [6], users can select a portion of the dataset to analyze (even the input data) and interact with the clustering results for validation and refinement. The users can interact with the clustering results by providing feedback [113] in the form of selecting, discarding, and fine-tuning cluster candidates [17].

3.2 Interacting with the Model's Parameters

From our collected papers, we commonly find interactions with the model (e.g., parameters). In particular, the idea is that once the initial clustering is performed and visualized, users are able to re-run the clustering using different parameters. But, contrary to the previous section, here, the individual users must decide which parameters to tune to improve the results. Surely, many of the proposed methods offer support to that end; however, influencing parameters are not automatically identified given the expected result from the users.

Conceptually, interactive visualization is used for helping the user to explore the space of alternative clustering results to refine the clusters by modifying the algorithm's parameters. The challenge here is that the user needs to determine which parameters to modify for a desired outcome–this step is not done automatically.

We find that most of the approaches support updating the usual parameters of the clustering methods, e.g., adjust the number of clusters or the similarity threshold parameters [5, 8, 19, 38, 43, 45, 49, 62, 65, 68, 70, 72, 79, 92–94, 101]. A somewhat unique perspective in this category is provided by Xiao and Dunham [111], where the authors note that when the whole dataset is unknown *a priori*, choosing an appropriate value of the number of clusters may be difficult. But fortunately, interactive feedback can be used to address this issue. Another common example is to identify parameters that make intuitive sense to the user and provide mechanisms that allow them to easily change, often through appropriate interface with visualization. Many of the solutions here are particularly adapted to a specific domain or application area. In topic modeling, several papers Bekkerman et al. [16], Choo et al. [30], El-Assady et al. [42] allow the user to select keywords for each topic, adjust their relative importance, and create new topics.

In another area, Wu et al. [110] proposes to use clustering to match input fields across two deep-web query interfaces. In this setting, they allow a human "integrator" to fix possible errors such as homonyms, synonyms, and do one-to-many mappings by splitting and merging clusters. In Nourashrafeddin et al. [77], the user can split and merge topics by assigning a term to other/multiple term cloud(s). Then, given the new term clusters, the algorithm re-clusters documents and provides improved cohesive topics (feature-supervised clustering).

In several cases, user feedback involves updating the weights, either different data instances or different features [34, 56–58]. For example, Babaee et al. [11] discovers the semantic structure of synthetic-aperture radar (SAR) image collections by letting users give higher weights to the most influential image in each cluster. This fits the general guidelines from Chuang and Hsu [32], which argues that the user must be provided with information regarding the contribution of input features toward the generated clusters and be able to modify the input features if necessary.

Several authors focus on simultaneous clustering using different parameters and giving the user tools to efficiently compare those results to find the best solution [42, 64, 71, 75, 105, 106]. For example, in Choo et al. [31], users can perform dimension reduction, interactively update the pre-processing options for data and clustering, apply different clustering algorithms, and visually compare the effects of these modifications across parallel coordinates, scatter plots, and cluster label views. In Zhang et al. [114], the visualization uses scatter plots, parallel coordinates, weights of individual dimension parameters, and different granularities at the same time. Highlighting one data item in one frame causes the same item to be highlighted in other views, enabling easy comparison between different settings. The user can also rearrange these frames by moving data with similar shapes closer together.

Alternatively, methods can focus on the exploratory aspect of interactive clustering, based on the assumption that the user is learning more and more about the data over time. In particular, this means that past decisions are not necessarily indicative of current goals. This is a justification for not interacting with the result and having the machine "guess" the user's intentions, but rather to be explicit about it. In this context, it is the sequence of (potentially different but related) clustering tasks that is of value, but this value is only known to the user. For example, in Brandt et al. [20], a user performs a sequence of clustering tasks using different subsets of data points and differently weighted attributes. For each of the sub-tasks, the user is expected to adapt model parameters taking visualization results into account from previous steps. As users learn more about the data, they not only specify better parameters for any given task but also narrow down their goals. In Bruneau and Otjacques [22], the system provides the data projection and its clustering simultaneously in the same two dimensional space, so that a user may influence the clustering output by directly manipulating the spectral clustering projection (e.g., a user draws a line that separates red and blue clusters, then the system re-clusters based on that line from the projection view).

Finally, in some cases, the focus of interaction is the data that is used for clustering. This is most common in geo-spatial data, where the user specifies an area of interest [19, 64]; document clustering, where the user can delete non-important data instances [67]; and biomolecular data [99]. Turkay et al. [99] proposes the interactive visualization method to reveal and analyze relationships in heterogeneous datasets at various abstraction levels (stratifications), aiming to find "elements of a cancer subtype that differ significantly from other subtypes." Exploratory interactive clustering allows analysts to match different intermediate results against other available information including clinical records or other meta-data. Alternatively, selecting which sub-part of data to use is decided in the context of large data, where clustering of all the data points would be too time consuming–in which case, data partitioning is done before the clustering starts [83, 108]. Other examples where interaction is done at the data level are in Andrienko and Andrienko [4], Dobrynin et al. [38], Hadlak et al. [50], Hund et al. [60], MacInnes et al. [72], Schreck et al. [86], Zhou et al. [116].

In many cases, the methods allow interactions with both the model and the results [7, 15, 19, 30, 32, 55, 79, 86, 94, 113]. For example, users can force desired meanings for a cluster by manipulating the weights assigned to particular terms within different topics, but they can also delete, merge, move, re-cluster, and sub-cluster the results. In another example, Schreck et al. [86] proposes an interactive visual system utilizing Self Organizing Maps (SOM), in which the user defines the trajectory or profile by monitoring the clustering process and controlling the progress at different detail levels.

3.3 Requesting Information from Users: Machine Initiative

Many papers in the interactive clustering field focus on presenting initial or preliminary results to the user and then giving them the freedom to guide the subsequent interactive process. However, there is also an active area where the opposite is true where the machine has the initiative and actively guides the user. In most cases, this is done through a series of queries where the algorithm identifies areas of insufficient knowledge and asks the user for clarification.

The most common queries are in the form of must-link/cannot-link constraints, often using ideas from active learning [107, 112]. For example, Vikram and Dasgupta [103] proposes an "interactive Bayesian algorithm that incorporates user interaction into hierarchical clustering" ([103], p. 1) and suggests several ways to intelligently query a user for constraints to be integrated into the clustering algorithm.

An alternative is to move data points between clusters. For example, in Iorio et al. [61], after each clustering iteration, the system suggests reassigning outliers to different clusters. The user then may accept or reject these suggestions and investigate alternative clusterings. In Dubey et al. [40], the algorithm requests users to assign data items to appropriate clusters and modify the corresponding feature vector to be semantically meaningful.

Instead of must-link/cannot-link constraints or moving data points, the system may also ask the users for true labels on the data, which works as a constraint. Sato and Iwayama [85] introduces an interactive, constrained, k-means clustering method on patent documents by iteratively assigning each item to the closest cluster and updating the corresponding centroids with the labeled documents.

A different approach is proposed in Mukhopadhyay et al. [74], where a system based on genetic algorithm optimization will interact with the user at pre-selected generations. As the solutions get better, the gap between two successive interactions increases and the rate of user intervention is reduced. As the user intervenes by ranking the top solutions, algorithm will de-emphasize those objective functions for the future that most disagree with the ranking.

4 INTERACTION OPERATIONS

In the reviewed papers, many different interactive operations were identified, such as the possibility to add, remove, and make corrections of the clusters generated. To classify these operations, we divided them in accordance with who initiated the operations, i.e., the user (Sections 4.1 to 4.3) or the clustering tool, i.e., the machine (Section 4.4). In addition, the operations were further divided into three sub-categories based on operations where the clustering tool aids the user to (1) visually explore, (2) change the number of clusters, and/or (3) make corrections of the clusters generated.

More details about the cited papers and its counts for Sections 4.1 to 4.3 are in the appendix (see Appendix A). A paper may contain multiple number of operations.

4.1 Visually Explore Clusters and Update

4.1.1 Compare Similarity. To assist the users in their cluster refinement operations, some tools present the results from various clustering settings/algorithms for the users to compare, aiding them to select the best configuration in accordance with some evaluation criteria (e.g., Choo et al. [31], Hadlak et al. [50], Okabe and Yamada [78]). For example, in El-Assady et al. [42], users are able to visually inspect the results from two different topic models at the same time where the interface presents the similarities and dissimilarities of the results of the two models. In Seo and Shneiderman [88], a user compares the results from two hierarchical clustering algorithms at the same time. Later in Seo and Shneiderman [89], a user can compare all possible pairs of sub-clusters with respect to precision, recall, and *F*-measure. In Choo et al. [31], users can adopt various clustering algorithms and review their results through the visualizations generated. In Mukhopadhyay et al. [74], the best results are presented to the users who rank the solutions according to their expertise. This ranking is then used to identify clustering quality measures that best match the given dataset and user.

As such, the comparison functionality can assist the users with various tasks–some tools enable the user to compare how the input data behave across different clusters (e.g., Okabe and Yamada [78], Turkay et al. [99]), whereas other tools focus on letting the users compare the results from different clustering algorithms [54, 65] or various parameter settings of a chosen algorithm [75]. Comparison functionality can also enable users to visually explore data points that fall into multiple groupings (e.g., Dudas et al. [41]).

4.1.2 Parallel Coordinate Interactions, e.g., Sort, Switch, Filter Feature on Axes. To aid the user in exploring generated clusters, some applications use parallel coordinate visualizations (e.g., Guo et al. [49], Hinneburg [52], Turkay et al. [99, 100]). In Lee et al. [67], the user can get an overview of the document, topic distributions, and the particular characteristics of a document through a parallel coordinates view. In their tool, a document line in the parallel coordinates view with many peaks marks that the document contains a mix of potentially related topics. In Kwon et al. [65], a user is able to sort and filter various cluster characteristics through the parallel coordinates visualization, whereas in Tatu et al. [98] and Guo et al. [49], interaction is for comparative tasks. In Zhang et al. [114], parallel coordinates view can be used to detect stable and unstable clusters by reviewing different parameter settings simultaneously.

4.2 Change the Number of Clusters

4.2.1 Change the Number of Clusters (or Other Parameters). Many of the papers reviewed include a function to change the number of clusters (e.g., Andrienko and Andrienko [4, 5], Babaee et al. [11], Basu et al. [15], Boudjeloud-Assala et al. [19], Brandt et al. [20], Chen and Liu [28, 29], do Nascimento and Eades [36], El-Assady et al. [42], Gaudin and Quigley [45], Guo et al. [49], Hoque and Carenini [53], Huang et al. [59], Jiang and Canny [62], Lee et al. [67], Lei et al. [68], MacInnes et al. [72], Packer et al. [79], Xiao and Dunham [111], Zhou et al. [116]). This functionality is often provided through the tools presented using various interactions and/or visualizations, such as sliders, node-link diagrams, matrix views, and dendrograms.

In some papers, the user receives help to estimate the optimal number of clusters to generate, such as in Kumpf et al. [64], where cluster split/merge diagrams and so-called "elbow plots" mark the optimal parameters. In Srivastava et al. [94], the user is presented with a first clustering of the data, and can then either reject, accept, or be ignorant of the result, triggering a re-clustering of the data. In Huang et al. [59], the user is assisted in this process by the application of the FastMap algorithm, which projects the clusters onto a 2D visual space where the user can inspect the cluster's compactness and isolation and, if needed, trigger a re-clustering process. In other papers, such as in Choo et al. [31], it is up to the user to select the most appropriate settings for different parameters, which is supported through simultaneous visualizations of the results from different clustering algorithms and settings.

Various interactions support this functionality such as adjusting the cluster threshold value [8, 65, 88, 92] by manipulating a contingency table that implements the scatter/gather approach [54]. Otherwise, Xu et al. [113] manipulates visualizations directly through the interface with adjacency matrices and treemaps where nodes can be removed or merged through dragging and dropping. Similarly, Guo et al. [48] use different visualizations (e.g., density plot, "subspace chooser," and high-dimensional clustering viewers) to cooperatively support the user in data exploration toward selection of the best clustering parameters, such as a distance threshold.

4.2.2 Split and Merge Clusters. It is quite common to split and merge clusters in many reviewed literatures, namely Cao et al. [24], do Nascimento and Eades [36], Dobrynin et al. [38], Erra et al. [43], Guo et al. [49], Lee et al. [67], Lei et al. [68], MacInnes et al. [72], Schreck et al. [86], Turkay et al. [99]. These operations are often prevalent in topic modeling applications, which allows a user to analyze and change the granularity of the topics generated (e.g., Choo et al. [30], Hoque and Carenini [53], Nourashrafeddin et al. [77]).

Split and merge operations are often performed directly through the interface. For instance, in Basu et al. [15], Chen and Liu [28, 29], desJardins et al. [35], Guo et al. [49], Turkay et al. [100], the user can perform drag-and-drop operations to edit clusters. But other interaction types are also available, such as through changing the parameters used in a contingency table or matrix (e.g., Balcan and Blum [13], Bruneau et al. [23], Hossain et al. [54], Looney [70]). Some tools, however, only support merging (e.g., Looney [70]), where the user is first presented with a large number of clusters that can either be removed or merged. Others only provide a splitting function. For instance, a user can split clusters by removing unwanted edges in a node-link diagram (Qiu and Li [81]) or a user can repeatedly split the clusters until only a pure "terminal" cluster remains (Huang et al. [59]).

To enable the users to perform good split/merge operations, visual support is often provided along with a tool that gives suggestions to split and merge. For instance, in Awasthi et al. [9], the algorithm performs local edits on the generated clusters based on user input. Moreover, in Awasthi and Zadeh [10], the user can request split and merge operations where the algorithm is instructed to merge two clusters if they have at least some fixed fraction of data points belonging to the same target cluster.

The most common visualization technique to support split/merge operations is a 2D scatter plot. However, other techniques have also been used, such as adjacency matrices and treemaps [113], where the user can remove nodes from one of the clusters or drag nodes into another cluster to merge them. In Arın et al. [8], cluster labels on a chord diagram are placed along the perimeter and selecting several labels allows the corresponding clusters to be merged.

4.2.3 Remove Clusters. Being able to remove clusters is a common function in the frameworks presented in Basu et al. [15], Chang et al. [27], desJardins et al. [35], Dobrynin et al. [38], Erra et al. [43], Guo et al. [49], Hadlak et al. [50], Lee et al. [67], Lei et al. [68], MacInnes et al. [72], Schreck et al. [86], Seo and Shneiderman [88], Srivastava et al. [94], Turkay et al. [100]. This operation is often enabled through drag-and-drop interaction in the interface, but other interactions are also allowed. For example, in Hossain et al. [54], a user can specify the inclusion/removal of clusters through a constraint table, whereas in Looney [70], a user can get rid of unnecessary clusters by setting a smaller number of clusters to be generated. In Liu et al. [69], a user can remove clusters that contain web search results that do not match the user's own semantic meaning of the query posed. This is done by manually labeling relevant clusters and then clicking on a "filter" button that removes unlabeled results.

4.2.4 Add Clusters. Many applications allow the user to add clusters, e.g., Basu et al. [15], Boudjeloud-Assala et al. [19], Chang et al. [27], either before the clustering process or after inspecting the results. This can be done by changing the number of clusters to be generated [64, 72] and by splitting and re-grouping the clusters [11, 40, 53, 67]. In Iorio et al. [61], a user can reassign outliers to different clusters after each clustering iteration.

4.3 Correction of Generated Clusters

4.3.1 Correction (of Error). Some clustering tools enable the user to make error corrections, for example, by changing the weights of the keywords, by selecting which documents to be used in topic modeling analysis [30], and by setting new evaluation criteria (e.g., potential cluster center (seed) and limits [19]). However, there are other approaches toward error correction. For example, in Srivastava et al. [94], the authors propose a "rejection-based approach toward interactive clustering," where a user can discard a given clustering and request another one. In Chang et al. [27], the user can correct the topic creation by "blacklisting" semantically erroneous topic labels by overriding the default number of clusters (by choosing "fewer," "normal," or "more," by directly editing the clusters, or by adding/removing clusters and topic labels). In PicHunter, the user corrects the system's "pick-representative-pictures" method by marking pictures representing the clusters as positive, negative, or neutral [76].

Several applications provide functions to move data points from one cluster to another, e.g., Basu et al. [15], Berthold et al. [17], desJardins et al. [35], Guo et al. [49], Hoque and Carenini [53], Lai et al. [66], Lee et al. [67], Nourashrafeddin et al. [77]. This can be performed by adjusting the centers and limits of the clusters, as in Boudjeloud-Assala et al. [19], MacInnes et al. [72]. In Sourina and Liu [93], the tool provides various geometrical shapes (e.g., boxes and ellipses) to update the clustering parameters. Others include dragging and dropping data points from one cluster to another [21, 33, 47, 66] and selecting data points that should/should not be in close proximity by interacting with the data points in a 2D projection [22, 23, 28, 29, 49].

4.3.2 Manipulate Features. In many previous studies, users were allowed to manipulate clustering features (e.g., remove, add, sort, and relocate). In Boudjeloud-Assala et al. [19], Kwon et al. [65], Lee et al. [67], Nourashrafeddin et al. [77], the user can remove data, features, and terms and decide which dataset dimensions to use for the clustering analysis. Then the algorithm reclusters using the given input. Brandt et al. [20] enables the user to choose a subset of features to use for each sub-problem as well as provide relative weights for those features. Siirtola [92] allows exploration with a limited rectangular selection within the data matrix, which thus contains a subset of instances or variables. Users can group clusters regarding attributes and filter them in Cao et al. [24]. More recently, in the INCREMENT approach (Mitchell [73]) selected instances from the sub-clusters are presented to a user for labeling to improve query efficiency. In studies concerned with clustering documents, individual words are considered to be features, and sometimes the user adapts the the word's weight based on the iterative results [56–58, 97]. In Bekkerman et al. [16], the users are kept active throughout the clustering process by selecting suitable features for the task-specific document representation. Furthermore, interaction in Corrêa et al. [34] allows one to extract a set of higher level features composed of correlated words relying on user's experience (e.g., "artificial," "neural," and "network" to "artificial neural network").

In addition, users can add features to the parallel coordinates [65] and sort features [92] in a reorderable matrix representing the data table, which functions as a visualization tool. Moreover, several tools allow multiple operations on cluster labels. In AppGrouper tool [27], the user edits the clusters directly, which includes adding/removing topic labels and editing pre-defined labeling of the clusters. In Dubey et al. [40], the user corrects misleading cluster descriptions by changing the cluster labels (so-called cluster description feedback). The clustering algorithm learns from this feedback and automatically re-clusters the dataset.

4.3.3 Make New Cluster with Constraints. In some papers, the user is allowed to create new clusters by constraints (e.g., Basu et al. [15], Choo et al. [30], desJardins et al. [35], Hu et al. [55], Lee et al. [67]). Some applications allow the user to modify the centroids (and limits) [19, 85], while others re-assign outliers to different clusters [61]. Hund et al. [60] allows the user to select a subset of interesting dimensions and include it in the clustering process.

The users can create their own clusters either directly through the visual representation [40, 65, 113] or by changing the values in a contingency table [54]. In Xu et al. [113], users can explicitly assign data points to a cluster and express their confidence in the cluster quality by directly manipulating the visual representations. In Dubey et al. [40], a user corrects misleading cluster descriptions by changing the cluster labeling, which the algorithm can learn from and use to automatically re-cluster the dataset.

In Gaudin and Quigley [45], the user can set the "similarity" parameters for a clustering, thus changing the constraints to be used in the algorithm. In Liu et al. [69], a user can further improve the clustering results by selecting which clusters to refine. The user can label the results and by clicking on a "filter" button, the snippets that do not belong to the label will be filtered out. Then, remaining results are re-ranked according to user's preferences. If the search results are unsatisfactory, then users can improve them by clicking on a "refine" button in the interface, which adds a selected cluster label to the original query.

4.3.4 Select Data to Cluster. Some clustering applications allow the user to select which data to cluster [6, 15, 19, 31, 49, 64, 65, 105, 106, 108]. In Kwon et al. [65], the user can set up constraints and force certain data points together/apart with different projection techniques (e.g., t-SNE, PCA, MDS) to help the user make a decision. In Wan et al. [105, 106], the user can iteratively cluster genome data using a combination of the "positional weight matrix (PWM)" [96] and k-means. Clusters with non-homogeneous patterns are re-clustered and the user can manipulate the window size to focus on detailed gene sequence. In Brandt et al. [20], the user can propose a sequence of clustering sub-problems, at each step selecting a subset of data according to the visualization of the previous iteration outcome. In Guo et al. [49] and Choo et al. [31], the user is able to select which data to cluster and make local changes of the generated clusters by selecting a particular region in a parallel coordinate plot or by selecting a different k-means seed in a 2D scatter plot. In Wang et al. [108], analysts can modify the data subdivisions through an interactive pixel chart view.

4.3.5 Add Constraints. Other forms of user-specified cluster constraints come in the form of so-called "must-links" and/or "cannot-links" (e.g., Ailon et al. [2], Basu et al. [15], desJardins et al. [35], Vikram and Dasgupta [103], Wang and Davidson [107], Xiong et al. [112]). These constraints define that two or more samples must, or must not, be assigned to the same cluster. The must/cannot links are specified when the user drags and drops images into new clusters [66], and in Wang and Davidson [107], violation of clustering is considered unsuitable, which may incur quality penalty. In Vikram and Dasgupta [103], the user specifies constraint triplets such as ({a,b},c), meaning "cluster should contain a and b but not c." In Geerts and Ndindi [46], the must/cannot links are specified by letting the user update a graph, resulting in deleting or re-labeling the edges until the user is satisfied. Most often, this functionality is provided directly through the user interface. For example, in Boudjeloud-Assala et al. [19], the user can specify constraints by placing data points closer/farther away from each other, whereas a user in Okabe and Yamada [78] must first select two data points and then specify a constraint type using a “must-link” or “cannot-link” button.

4.3.6 Compare and Correct/Validate. There are many studies that allow users to compare and then correct, e.g., Castellanos-Garzón et al. [26], Hinneburg [52], Huang et al. [59], L’Yi et al. [71], Turkay et al. [100]. In fact, visualizing with appropriate representations and allowing iterative interactions are crucial for better interpretation and validation. In Tatu et al. [98], the user iteratively decides parts of relevant subspaces (e.g., list of 2D scatterplots with a corresponding parallel coordinates plot) sorted by the interestingness index from the clustering algorithm with the help of a multi-dimensional scatter plot. In L’Yi et al. [71], users can modify the input parameters of the multiple clustering techniques after comparing clustering results. In Okabe and Yamada [78], the user can compare clusters before and after adding constraints to select the best must- and cannot-link constraints.

4.3.7 Pick the Appropriate Parameter Setting by Comparing Visualizations. By inspecting the results from two or more algorithms and manipulating parameter settings, the user can more easily compare clusters and select the most suitable model. There are tools that allow users to compare the clustering results through visualizations and pick the appropriate parameter during the clustering process (e.g., Choo et al. [31], El-Assady et al. [42], Müller et al. [75], Zhang et al. [114]). User’s interactive approach applies similarly with topic modeling, because it keeps the best performing algorithm as an anchor for the next iteration [42] and to subspace algorithms [75].

Moreover, a tool can provide preliminary feedback during the training period by letting users control multiple parameters [62]. Users can select a model to run (e.g., k-means, NMF [Nonnegative Matrix Factorization], LDA for topic modeling), parameters to change (e.g., size weight, batch size, rate), and evaluation metrics to use (e.g., silhouette graph, similarity matrix, cluster size balance in a histogram). Domain experts can also assign higher weights to particular input images using k-means when they already have knowledge of the input data [11].

4.4 Machine Initiates

There are only a handful of approaches where the machine initiates the interaction and several of them are rather simple. The lack of more advanced work in this area is quite surprising.

The clusters are available in the order by cluster size [8], by time [100], and by relevance [98]. The I-TWEC tool in Arın et al. [8] uses bar charts to represent clusters sorted by size. Turkay et al. [100] provides a temporal view of clusters and minimizes the overlap of the connections between clusters. Tatu et al. [98] allows users to select a subset of relevant subspaces for a more detailed comparison based on interesting patterns observed in the subspace view.

In some cases, the algorithm removes clusters. In Dubey et al. [40], Kumpf et al. [64], the tools provide the results of splitting, merging, and re-clustering based on the user’s feedback and decision. Kumpf et al. [64] presents both the possible splits and merges of clusters and how a change may affect the robustness of the clustering results.

Some tools can suggest corrective measures when an error is suspected. For example, users are encouraged to select documents that are closer to a certain topic [42], fix which item belongs to which cluster [15], accept or reject clusters [94], and focus on where to perform local updates [32]. Chuang and Hsu [32] emphasizes that clustering operations should be only applied to relevant data. In El-Assady et al. [42], user feedback from a recommendation significantly influences the next learning iteration to create an ideal topic composition from ambiguous documents.

5 INCORPORATING THE USER'S FEEDBACK

The interactive clustering process is a loop in which the clustering algorithm continuously communicates information to users and takes feedback from them. Closing the loop is an important step in this interactive process and requires a mechanism to incorporate the user's feedback. This is done in various ways in different papers. Most of the papers suggest directly modifying the cluster's structure (Section 5.1), adjusting similarity functions or parameters of the clustering model (Section 5.2), or interpreting feedback as a constraint in a constrained-clustering algorithm (Section 5.3). Other alternatives include simple strategies such as accepting/rejecting a proposed clustering solution and re-clustering (Section 5.4).

5.1 Modifying the Cluster's Structure

The split and merge commands from a user can directly modify the structure of clusters [10, 13, 43, 66, 67] without any specification of how a cluster should be split. In Choo et al. [30] and Hoque and Carenini [53], the users incorporate their domain knowledge through splitting and merging topics (modeled as clusters) according to their preferences. In Looney [70], users create their feedback by iteratively adjusting the value of a merging parameter so that small clusters are removed and similar clusters are merged. In Iorio et al. [61], instead of formulating a split request, the user can choose to reassign outliers to different clusters after each iteration.

A cluster's structure can also be changed by updating the number of clusters. For example, in Chang et al. [27], users can steer the algorithm by indicating preferences for the number of clusters to obtain more coarse or fine-grained clusters. In Xiao and Dunham [111], user's feedback consists of incrementally increasing or decreasing the number of clusters in agglomerative hierarchical clustering. After the number of clusters is changed, the modified dendrogram reflects the new number of clusters. In Choo et al. [31] and Sourina and Liu [93], users interact with the interface to change parameters (including the number of clusters) that affect clusters' geometrical shapes.

In some cases, the user directly modifies the underlying clustering model. For example, in Qiu and Li [81], the clustering model is represented as a graph structure and the user is allowed to remove unnecessary edges between data points that should belong to different clusters. In other papers, modifying the structure of clusters is done by interacting directly with a visual interface. For example, in Zhang et al. [114], the user can edit the scatterplot visualizations, moving data with similar shapes close to each other. In Guo et al. [49], the user can add attractive/repulsive operators on the axes of a parallel coordinate plot and make local changes to clusters in a specific region. The model uses these operators as user-specified parameter strengths. In the same context, users in Berthold et al. [17] can select, discard, and fine-tune cluster candidates through a simple visual interface that displays distances to cluster's centroids.

5.2 Adjusting Similarity Functions and Parameters

Some approaches incorporate user's feedback by adjusting a distance (dissimilarity) matrix, adjusting similarity and dissimilarity functions, and assigning weights to the features and readjusting these weights iteratively. For example, in Choo et al. [30], El-Assady et al. [42], Lee et al. [67], users incorporate their domain knowledge by adjusting the weights of keywords used as features. In Coden et al. [33], a user can move points from one cluster to another and with each move, the method adjusts weights used in the weighted semantic similarity measure. In Okabe and Yamada [78], the interactive clustering tool repeatedly uses "distance metric learning" to update a dissimilarity matrix to satisfy user's constraints through the interactive process. Similarly, the primary method for propagating user's suggested changes [23] is to update the dissimilarity matrix. The method presented in Mitchell [73] selects instances from sub-clusters and presents them to the user for labeling. Any feedback given will update the clustering model so that spatial distance is inversely correlated with semantic similarity based on the labels.

Regarding to clustering text documents, the methods in Hu et al. [56–58] present a subset of top-ranked features to the user. A user can tag features as "accepted" if they are believed to be useful for discriminating clusters for every iteration. All accepted features, together with a few of the top-ranked remaining features, are included in the next clustering iteration. This new feature space, with accepted features given higher weights, are then used to re-cluster the documents. In the same context, in Corrêa et al. [34], a user's feedback for re-clustering the data is composed of high-level textual features (composed words) combined with existing word-level features.

Other papers incorporate the user's feedback by adjusting parameters instead of similarity functions. An example of this is presented in Lei et al. [68], where weight settings in the clustering algorithm are adjusted to reflect the user's preference as much as possible. In Jiang and Canny [62], users can investigate tradeoffs among competing goals made by different parameter choices and visualize their effect on clustering results. In Zhang et al. [114], the user can change the parameters used by the clustering algorithms by adjusting dimensionality and granularity settings. Similarly, in do Nascimento and Eades [36], user feedback is incorporated by letting a user set the parameters, such as number of clusters, material properties, regions of interest. Then, the tool re-runs the clustering algorithm and visualizes the result so that a user decides to tweak the parameter settings further. Another example is Andrienko et al. [7], where the user is able to change distance thresholds for sub-cluster prototypes, which will trigger a re-computation of sub-cluster's medoids. Mukhopadhyay et al. [74] proposes the interactive multi-objective clustering (IMOC) algorithm, which periodically learns from the user which validity measures are a better fit for the dataset being clustered. As the user ranks top clustering solutions higher, validity measures that do not conform to the ranking are penalized.

5.3 Using a Constrained-clustering Algorithm

In some papers, user's feedback constrains the clustering algorithm. Most of these constraints are expressed as "must-link" or "cannot-link" between data points' pairs, as in Basu et al. [15], desJardins et al. [35], Hu et al. [55], Wang and Davidson [107], Xiong et al. [112]. However, it can be expressed differently, as in Basu et al. [15] where a data-point is constrained as "must-belong" or "cannot-belong" to a cluster. The user can either formulate and impose these constraints on the clusters [15, 35, 55] or answer a query (formulated by the algorithm) with such constraints (e.g., "should a and b be linked?") [107, 112]. The constraints can also be incorporated by allowing the user to label relevant clusters and filter out the clusters that do not match the users' view of the clusters [69].

These constraints can be incorporated in a "distance metric learning," as done in Basu et al. [15] or by directly modifying the objective function for spectral clustering [107, 112]. But such constraints can be more naturally incorporated in a constrained clustering algorithm. For example, in Sato and Iwayama [85], constraints are imposed by assigning true labels to the clusters and applying a constrained k-means algorithm. Dubey et al. [40] allows two types of feedback: the "assignment feedback," where the user reassigns a data point to a different cluster, and the "cluster description feedback," where the user changes the cluster feature vector to make it more understandable and meaningful. The authors interpret these two kinds of feedback in terms of constraints, incorporating them into assignment and update steps to balance the distortion error against the constraint violation penalty. Similarly, desJardins et al. [35] integrates "must-link" and "cannot-link" restrictions into PCKmeans (Pairwise Constrained K-means) algorithm [14]. In Vikram and Dasgupta [103], hierarchical clustering is used and the algorithm queries the user for constraints about a specific sub-tree. The user's feedback consists of answering the query with

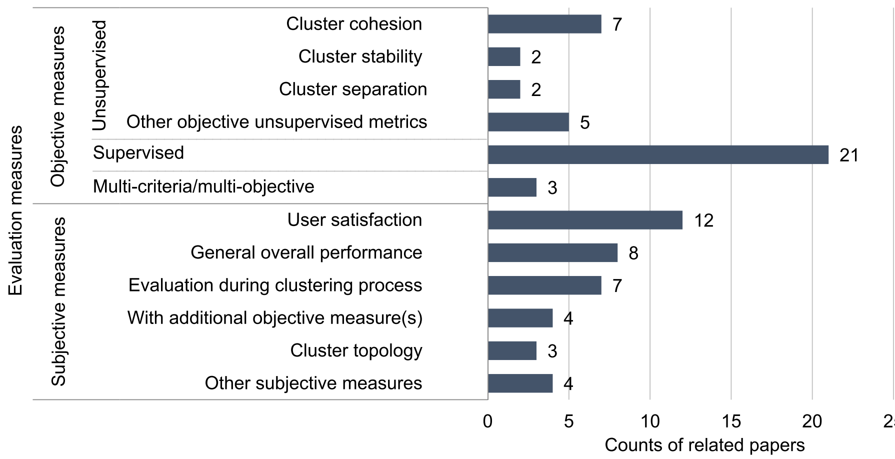

Fig. 1. Summarized categorization of evaluation measures with counts of reviewed papers.

constraints of the ({a, b}, c) kind—meaning a group should have a sub-cluster with *a* and *b* but not *c*. These constraints are then incorporated in the hierarchical clustering by modifying the tree so that the constraints are satisfied. Another example includes Gaudin and Quigley [45], where the user can set "similarity" parameters to be used as constraints; clustering is then performed iteratively until the user accepts the clustered node-link diagram.

5.4 Others

Other papers propose simple feedback. For example, a user can accept or reject proposed clustering solutions or mark whether a picture is correctly clustered or not [76, 94]. In other cases, a user can adjust a threshold [7, 8, 48], delete noisy features [16], adapt the weights [97], and reassign outliers to different clusters [61].

6 EVALUATION MEASURES

The evaluation of interactive clustering methods and systems is an open challenge, highlighted by several researchers within interactive machine learning [3, 117]. Machine learning research field has well-known and accepted conventional metrics to evaluate performance such as precision, recall, accuracy, squared error, likelihood, posterior probability, information gain, and so on (measures to evaluate clustering results are briefly reviewed in Fahad et al. [44]).

Evaluations found in the reviewed papers can be classified as objective and/or subjective measures. In turn, objective measures are divided into unsupervised and supervised measures (see Figure 1). Objective unsupervised quality metrics include, among others, cluster cohesion, cluster stability, and cluster separation, while objective supervised metrics are used when some form of ground truth is assumed (e.g., accuracy, recall, F-score). Subjective quality measures normally involve evaluations such as user satisfaction and joint performance. Figure 1 presents the number of collected papers we reviewed for each category. In the following sections, we present a comprehensive summary of the most relevant metric used during the evaluation of interactive clustering methods and systems following the classification proposed.

6.1 Objective Evaluation Measures

6.1.1 Unsupervised Metrics.

Cluster cohesion. Cluster cohesion is a measure of how closely objects are related in a cluster. This is a quite common metric used for clustering evaluation [8, 19, 27, 61, 73, 100, 112]. Iorio et al.

[61] uses average stability as an evaluation criteria for the resulting clustering, while Turkay et al. [100] employs a vicinity measure to the compactness (cohesion) of clusters. The process ends when the user decides to stop or no more clusters can be merged. In Boudjeloud-Assala et al. [19], a user is aided in cluster evaluation through automatic measures of how compact the various clusters are. Mitchell [73] measures cluster homogeneity. In the I-TWEC system [8], the interactive merging phase lets users combine clusters based on semantics, i.e., perceived relatedness of particular set of words and the cluster labels content.

Cluster stability. Cluster stability is based on the idea that a structure of interest in any domain needs to be "durable," i.e., such structures must be similar if they are created from various sets of data generated from a single underlying process or based on a single model [104]. We find Awasthi et al. [9] and Sato and Iwayama [85] use this type of metric. Awasthi et al. [9] uses a stability property–a generalization of "stable marriage" property that works on real-world data. Sato and Iwayama [85] introduce a way to predict and measure accuracy (defined as the number of properly classified documents divided by corpus size) based on cluster stability (e.g., the number of changes of cluster labels).

Cluster separation. Cluster separation assesses how distinct clusters are from each other. In Boudjeloud-Assala et al. [19], Chang et al. [27], a user evaluates the cluster by applying penalty criteria to overlapping clusters.

Other objective unsupervised metrics. A number of the reviewed papers used other objective, unsupervised metrics when labels are not available. Such standards are normally found in data mining and machine learning evaluations. For instance, for clustering- and classification-related tasks, the following metrics can be used: variance-based methods, a silhouette index, homogeneity, compactness, a Davies-Bouldin index, rate-distortion methods, and stochastic complexity. Rawlins et al. [83] uses two tests (the Kolmogorov-Smirnov 2-sample test and the Mann-Whitney U-test) to compare cluster distinctions. Qiu and Li [81] uses evaluation metrics not discussed in detail but described as "compact (small distance within clusters) and distinguishable (large distance between clusters)." Many authors, for example Wang et al. [108], compare the quality of their clustering results with standard k-means. In Liu et al. [69], the clusters generated are evaluated by employing objective measures such as cluster compactness, phrase length, and term frequency (Inverse Document Frequency); cluster evaluation is complemented with subjective user feedback.

Another objective unsupervised metric is based on clean-split assessments (Awasthi et al. [9]). The authors evaluate the effectiveness of the split process by computing a "Clean-Split." This proposal can be compared to other binary splits, such as 2-median clustering and spectral-balancing. The authors found that their new metric computes the best splits; accordingly, they evaluated the computed splits using the correlation-clustering error.

6.1.2 Supervised Metrics. In this category, we find metrics that use labeled data or some assumed ground truth to compute clustering performance. Examples of this are accuracy, normalized mutual information (NMI) index, F-score, recall, adjusted Rand index, and information-theoretical index. There are numerous works that employ one or several of these metrics in an evaluation. We will now provide a summary of such studies.

Both accuracy and normalized mutual information (NMI) appear very often as evaluation metrics. For instance, Babaee et al. [11] uses accuracy (percentage of correct labels based on the prior annotations) and NMI (0–1, which decides whether two clusters are dependent or independent); Hu et al. [56–58] uses accuracy and NMI as well. In Andrienko et al. [7], the ratio of false negatives and false positives is used in addition to the subjective assessment for evaluating cluster cohesion. Wu et al. [110] and Bekkerman et al. [16] evaluate the clustering performance not only through accuracy and NMI but also through adding recall, precision, and the F-score. Those last three are also presented in Seo and Shneiderman [89] for a user to compare all possible sub-cluster pairs. NMI and F-score are as well used in Corrêa et al. [34] and Nourashrafeddin et al. [77]. Nourashrafeddin et al. [78] evaluates clusters through true documents' labels and a confusion matrix, which is used to compute F-scores and NMI. Another example of utilizing NMI is Okabe and Yamada [78], where the score is used to evaluate the effectiveness of the constraints, allowing comparison between fully automatic and interactive clustering process.

Rand index is employed by Wang and Davidson [107] to evaluate clustering results. Homogeneity of clusters is measured in Turkay et al. [100] with two standard deviation measures. Looney [70] uses a modified Xie-Beni validity measure to assess compactness and separation among clusters. Erra et al. [43] performs a quality test on their method using precision and recall. They find that precision and recall values were higher than those compared with k-means and hierarchical approaches. The VC+ tool of Huang et al. [59] evaluates clusters based on a measure of dominance, i.e., a measure built on the frequency of categories within a cluster. In Lai et al. [66], the first unsupervised clustering result is evaluated by the Silhouette-Width quality measure and the outcomes are analyzed considering the ground truth using the V-measure; thereafter, it is up to the user to refine the clusters by adding must-link/cannot-link constraints. "Evaluation of the Increment" approach presented in Mitchell [73] is based on the accuracy, homogeneity, completeness, V-measure, and relative Jacquard coefficient of the clustering. In Jiang and Canny [62], evaluation metrics are silhouette curves, similarity matrix, cluster size balance in a histogram, a likelihood graph (k-means), series of recovered images (Non-negative Matrix Factorization), and topic-term matrix visualization with a likelihood graph (topic modeling using LDA [18]). The AppGrouper tool [27] is evaluated by granularity and coverage of clusters and balance of cluster sizes. The I-TWEC tool [8] uses intra-cluster similarity and cluster purity to evaluate the results.

6.1.3 Multi-criteria/Multi-objective Clustering. Some evaluation solutions include multi-criteria and multi-objective measures. For example, Mukhopadhyay et al. [74] uses multi-criteria optimization by combining five indexes: DB, Xie-Beni, J_m, PBM, and Silhouette. AIDE assistant [95] uses several standard metrics to evaluate clustering results (however, the authors do not provide details). Clusters in Kwon et al. [65] are assessed using quality metrics such as Silhouette, Calinski-Harabaz, SDbw, Gap statistic, and Davies-Bouldin. They visualize the quality metric scores using radar charts. The features are ranked by F-statistic, which orders a feature's significance and retrieves the associated p-value. Statistical summaries are provided in bar charts with cohesion, separation, and silhouette scores.

6.2 Subjective Evaluation Measures

In interactive clustering, a user is involved throughout the process, and, as argued in Chuang and Hsu [32], users should carry out the evaluation. This is seen in several reviewed papers that aim at assessing clustering results from experts and users. User evaluation is presented in e.g., Andrienko and Andrienko [4], Andrienko et al. [6, 7], Arın et al. [8], Berthold et al. [17], Boudjeloud-Assala et al. [19], Cao et al. [24], desJardins et al. [35], Dobrynin et al. [38], El-Assady et al. [42], Guo et al. [48], Hadlak et al. [50], Hoque and Carenini [53], Hossain et al. [54], Hund et al. [60], Lei et al. [68], MacInnes et al. [72], Sato and Iwayama [85], Zhou et al. [115, 116]. Mainly, these works present visual and interactive solutions to show the clustering process and results. They primarily support refinement solutions until the user is satisfied with them. Here we provide a summary of works that use subjective measures.

6.2.1 Based on User Satisfaction with the Clustering Results. Chen and Liu [28, 29] focus on using a visualization framework called VISTA to help users validate and refine clustering results visually. Brandt et al. [20] allows users to iteratively define clustering sub-problems until they are satisfied with the results, which is similar to the approach in desJardins et al. [35], where users decide if clusters are suitable.

In Choo et al. [30], the user analyzes the topics generated and evaluates them, which is a method also used in Boudjeloud-Assala et al. [19], where the analyst makes subjective evaluations of generated clusters and adjusts them accordingly. Similarly, Chang et al. [27] uses expert decisions to guarantee that the resulting topics are coherent, their labels are semantically meaningful, and the granularity of clusters is appropriate as a decision maker optimizes the tradeoffs between these requirements. Moreover, in the topic-modeling domain, it is very common for the user to determine how well topic models perform on the corpora used. In Takama and Tonegawa [97], an evaluation of user's satisfaction is based on the combination of surveys as well as the number of different topics discovered within the document corpus. The work presented in Lai et al. [66] describes how a user can iteratively evaluate clustering results by inspecting images contained in each cluster. Then, the user drags and drops incorrectly clustered images into appropriate clusters. Gaudin and Quigley [45] allows the user to determine when a clustering is satisfactory by reviewing the node-link and matrix views of the social network data. Along the same lines, in do Nascimento and Eades [36], it is the user who establishes when clustering is complete by reviewing how well the results fulfill any constraints set by the user. To review how well the algorithm meets the constraints, three different visualizations are offered: a histogram to view how many clusters there are by size, a scatterplot to show the size of each cluster, and a node-link graph that shows the current feasible clustering given the constraints. The user can then decide if the clustering is satisfactory, if additional parameter tweaking is needed, or if a different clustering algorithm is needed. A combination of objective and subjective evaluations is carried out in Liu et al. [69], where clusters are evaluated for objective measures such as cluster compactness, phrase length, and term frequency (inverse document frequency but also subjective user's opinions).

6.2.2 General Overall Performance. An interesting and somehow unique user evaluation is presented in MacInnes et al. [72], where it shows that final clusters would not have been obtained using solutions that are either solely automatic or solely human (or user) based. The VC+ tool of Huang et al. [59] visualizes several cluster characteristics including cluster compactness and relationships, distribution of classes in clusters, and cluster isolation. Using the visual representations of a cluster, a user can assess how many sub-clusters there are (if any), potentially triggering a new clustering iteration. Moreover, in Andrienko and Andrienko [5], the clustering results (temporal patterns) correspond to the authors' background knowledge of user activities, while in Muller et al. [76], a preliminary evaluation of the PicHunter tool is performed by a user who tries to find the target cluster.

The user evaluation held with the tool ConVisIT Hoque and Carenini [53] compares two other tools to revise topic models by finding interfaces and interactions that are more helpful to interpret cluster data. The work presented in Guo et al. [49] also includes a user study, which evaluates whether the analysts could, with their tool's help, identify hidden clusters in the data with high accuracy. They found that participants who interacted most with the tool (i.e., made the most changes in terms of attractive/repulsive operations) were the ones who found the most accurate clusters (the mean accuracy was shown to be about 90% for each identified cluster). This means that the interaction helped users to better understand the data and find hidden patterns in it. The user has the choice to perform clustering manually or automatically in Zhang et al. [114]. If manual clustering is opted, then the user decides when a clustering is satisfactory. With automatic means, the authors use a compactness measure (external/internal connecting distance). Finally, in Sourina and Liu [93], it is once again the user who determines when a clustering is complete by inspecting the visuals generated by the tool.

6.2.3 User Evaluation During the Clustering Process. In Zhou et al. [115], clusters of locations are provided on a map and the user selects locations deemed interesting. Subjective assessments of how well the presentation supports this particular task is the basis for the clustering evaluation. Another approach, focusing on labeling, is shown in Sato and Iwayama [85], where the user is allowed to cluster patent documents by updating the true labels of clusters.

User involvement in the clustering process and evaluation is presented in Dubey et al. [40]. In each iteration of the algorithm, after minimizing the total clustering error while considering all past feedback, the results are inspected by a user. New feedback is given by browsing through cluster definitions and checking how objects have been re-assigned. Hossain et al. [54] proposes an interactive clustering process where a user inspects clustering results, criticizes them, and proposes new parameter settings (e.g., to modify the number of clusters). This triggers running the clustering algorithm again to obtain a new clustering result. In the I-TWEC tool [8], after the resulting clusters have been presented using histograms, bubble charts, and chord diagrams (wheel charts), the end-user can examine the results and adjust the clustering threshold. Again, it triggers a re-clustering of the dataset. In the TINDER approach of Srivastava et al. [94], a standard algorithm initially clusters the data and user inspects the presented clustering results. If that result is not satisfactory, then the user can provide per-cluster feedback (reject, accept, or do neither). This way, a more directed style of exploration is accomplished, in which users guide the clustering procedure toward a partitioning that interests them. In L'Yi et al. [71], domain experts subjectively evaluate the usefulness of visual clusterings produced by XCluSim.

6.2.4 User Evaluation with Additional Objective Measures. There are a few examples that combine subjective evaluations and objective metrics. For instance, in Andrienko et al. [7], a user refines clustering of trajectories (or routes) if the internal variation is assessed too high measured by low level of cluster cohesion. Brown et al. [21] uses 10 college students to find that all user-guided distance functions result in more cohesive clusters than just using the euclidean distance function. Castellanos-Garzón et al. [26] presents a combined evaluation where both objective and subjective measures are used. The aim was to demonstrate the consistency of the numerical and visual validations against a reference clustering. The objective measures utilized are the Rand index, Jaccard coefficient, Minkowski measure, and Adjusted Rand index. Finally, the authors from Huang et al. [59] use a mixture of objective measures (accuracy of the classification, comparing their suggested algorithm with other decision-tree algorithms) and subjective user evaluations.

6.2.5 Cluster Topology. A particular class of subjective measures refers to assessing the topology of the generated clusters by the users. For instance, in Hossain et al. [54], expert users visually inspect alternative clustering results to find the alternative that gives the "clearest" borders between classes. Tatu et al. [98] utilizes a linearly sorted view of subspaces (list of 2D scatterplots with a corresponding parallel coordinates plot) from which the user can select a small number of relevant subspaces for a more detailed comparison based on interesting patterns observed in different views (e.g., point distribution, point density, and stripes). They improve understandability of the clustering topology by using a "multi-dimensional scaling layout of the total number of subspaces with cross-colored group representatives" ([98], p. 8). The subjective evaluation carried out in Packer et al. [79] supports analysts (not domain users) who are assessing and trying to gain insights into cluster topology and parameters.

6.2.6 Other Subjective Measures. Here, we include works that use agreement from the users, number of queries to the user, and measures based on interpretability and explainability.

User agreement. Subjective measures may also include user agreement for a particular task. For example, Coden et al. [33] uses the Adjusted Rank Index between groupings for several subjective tasks from five users (e.g., categorizing items on a restaurant menu). Another example is presented in Wu et al. [110], which the goal is to reduce uncertainties during the matching process by asking users whether two chosen synonym or homonym fields match (yes or no answers).

Number of queries. Several papers present "active clustering" methods that formulate queries for the user. In this context, interactive clustering algorithms are query-efficient if they involve only a few interactions with the user. In Vikram and Dasgupta [103], algorithm queries the user for constraints and the evaluation takes into account the number of answers required to obtain a clustering similar to the reference one.

Based on explainability and interpretability. Bruneau et al. [23] demonstrates an extensive evaluation of their approach on several datasets that focus on the explainability and interpretability of the results.

7 CLUSTERING WITH INTERACTIONS (DISTINGUISHED FROM INTERACTIVE CLUSTERING)

This section focuses on helping a user to inspect and understand the clustering results without providing explicit mechanism for actual "interaction." We have considered whether some of the selected papers are within the scope of this survey and we include them so that we have a broader perspective.

7.1 No Feedback Reflected to the System to Update Clusterings

This section contains approaches that primarily aim to support the user in inspecting and understanding the clustering results, without providing any explicit mechanism for actual *interaction*—systems that only visualize clustering results. In most cases, of course, these visualization tools are interactive (allowing operations such as zooming). We have actually considered whether these papers are within the scope of this survey and decided to include them, because it is often not immediately obvious neither from titles nor from abstracts. This clearly indicates that there is a need to establish more clear conventions and nomenclature within the field—we believe that in a survey like this, it is useful to demonstrate how serious the issue is.

A representative example is St. Amant and Cohen [95], which describes an AI-based Exploratory Data Analysis assistant (AIDE) that can perform several different data analysis tasks, including clustering. AIDE has a certain degree of autonomy in that it can perform an initial analysis of the data independently. It then presents the most interesting outcomes to the user, allowing them to focus the analysis on their personal areas of interest. There is no interactivity in the clustering step, though—the user, after receiving the results, can request a re-run of the clustering with different parameters or can ask for an explanation of what has prompted AIDE to select a particular method, but the user cannot interact with the clustering process itself.

In some cases, the interactivity is justified by computational requirements—any visualization that takes too much time and computational power to generate is more or less useless in an interactive setting, thus methods to speed up the operation (through optimization, approximation, or pre-fetching results) fit here. For example, Ahmed and Weaver [1] describes a heuristic pre-fetching method that leads to improved response time of the interactive k-means algorithm in "dynamic query visualizations of multidimensional data" ([1], p. 1). Rasmussen and Karypis [82] provides a clustering platform that allows running multiple clustering algorithms with a work-flow, i.e., import and prepare data, select clustering options, generate reports, and display visualization.

In certain domains, when the clustering is a means to an end, the interaction may not be directly with the clustering itself but rather with the results. For example, Zhou et al. [115] uses deterministic clustering to identify a user's significant places from their location data (assuming that clusters of GPS positions correspond to interesting locations). It lets the user to choose either accept or reject identified locations.

In most cases, the core contribution of the papers within this category is the visualization itself. For example, in Guo et al. [49], high-dimensional data are visualized using parallel coordinates. The tool applies attractive and repulsive operators to regions of interest using an electricity interaction metaphor. Van Long and Linsen [102] proposes visualization and interaction approaches with a focus on individual clusters, as well as methods to distribute sub-clusters within and across cluster hierarchy levels. This work is continued in Dobrev et al. [37] with a specific focus on spatial data by using clusters to interactively assign material properties (for example, opacity or color) that are directly mapped to a transfer function. This transfer function is applied to render the multivariate volume data inside the linked 3D texture-based rendering view. In Seo and Shneiderman [88], the tool provides four features of the hierarchical clustering results: overview-detail view, dynamic query controls to highlight the interesting clusters, coordination of overview and the scattergrams, and cluster comparisons. For temporal data, Turkay et al. [100] presents two interactive visualization techniques—called "temporal cluster view" ([100], p. 5) and "temporal signatures" ([100], p. 6). These enable users to perform identification and interpretation of how clusters change in terms of their temporal structures. In Chen and Liu [28, 29], the authors argue that interactivity is needed to find clusters of irregular and complex shapes. They show the limitation of cluster validity methods used on arbitrarily shaped clusters and it is virtually impossible to determine the optimal cluster structure automatically. Since irregularities generally cannot be anticipated, statistical methods are sufficient, and visualization frameworks lead to better cluster structure. Dudas et al. [41] presents a visualization that models overlapping community memberships in large network topologies using glyphs. In Qiu and Li [81], the authors propose a tree-based method for representing the data, which allows clear visualization of the clustering, especially the unwanted edges. In Sarnovsky [84], the user can visually inspect models constructed with Growing Hierarchical Self-Organizing Maps (GHSOM) by browsing through the model structure to explore different map layers and even particular neurons and their content. In Cao et al. [25], node-link diagrams are used to explore the relations between text documents on local and global levels.

Such visualizations are often geared toward understanding clustering results through the simultaneous use of several synchronized views. In Seo and Shneiderman [89], it is a combination of dendrograms, heat maps, and scatterplots; in Takama and Tonegawa [97], it is information at the "cluster level, document level, keyword-set level, and keyword level" ([97], p. 1); in Tatu et al. [98], the visualization includes scatterplots, similarity-based list views, and parallel coordinates. Guo et al. [48] argues the importance on spatial dimensions because of its complexity (e.g., geographic obstacles) and difficulty to reflect the real-world measures (e.g., road distance).

In Guo et al. [48], the authors focus on spatial clustering, arguing that "spatial dimensions cannot simply be treated as two additional non-spatial dimensions in general clustering methods" ([48], p. 2). The very special meaning that those spatial dimensions exhibit in the real-world directly affects the types of structures the user is interested in finding—and those structures are almost impossible for general-purpose clustering methods to find. Moreover, the distance or similarity measures within spatial dimensions often need to take into account complex contextual dependencies; examples include road networks, administrative or political regions, and natural obstacles such as rivers or mountains. Therefore, the work focuses on supporting the user in integrating spatial dimensions into larger general-purpose feature spaces in a way that preserves this special meaning, so that high-dimensional clusters can be identified in the resulting combined attribute space in an interactive fashion.

7.2 Clustering with Interaction Operations

7.2.1 User Initiates: Hide and Expand Clusters. Many of the reviewed articles describe allowing users to hide and expand the clustering result (see Andrienko and Andrienko [5], Andrienko et al. [7], Basu et al. [15], Boudjeloud-Assala et al. [19], Dobrynin et al. [38], Hadlak et al. [50], Lee et al. [67], Schreck et al. [86], Seo and Shneiderman [88], Zhang et al. [114]). For example, Hadlak et al. [50] provides such functionality using different visualizations for dynamic networks to discover substructures within the network that share similar trends over time. Rasmussen and Karypis [82] presents a tool for analyzing large sets of genome data, allowing the user to inspect both global and local structures of the data through hierarchical clustering and dendrograms, thereby reducing visual clutter. Similarly, Hoque and Carenini [53] allows the user to visually explore the chronological development of conversational data with an overview+details analysis of how conversations have evolved over time.

Other papers provide such functionality to enable the users to find different clusterings in high-dimensional data. For example, in Tatu et al. [98], the user can expand each subspace clustering from a preceding step, allowing the individual subspaces to be inspected and compared in detail. In Hossain et al. [54], the user can choose which of several alternative clusterings should be visualized and used as bases for further analysis. In Sarnovsky [84], the overall hierarchical structure of the GHSOM model can be investigated using the zoom in and out functions, and the user can expand and prune the map layers using a combination of data tables and dendrograms.

7.2.2 User Initiates: Zoom in and Out. To visually compare the similarity of different clusters, many tools provide zoom in and out functions, for example, Boudjeloud-Assala et al. [19], Cao et al. [24], Choo et al. [31], Hadlak et al. [50], Hoque and Carenini [53], Lee et al. [67], Seo and Shneiderman [88], Tatu et al. [98], Xu et al. [113], which many attempts to solve the scalability issue in clustering. Shneiderman's information-seeking mantra [90], "overview first, zoom and filter, then details-on-demand," forms the basis for the visual analysis of alternative clusterings.

Boudjeloud-Assala et al. [19] and Tatu et al. [98] compare similarity of subspaces using zooming in and out. A subset of data objects are processed by selecting point of interest of the data based on user's interest. Hadlak et al. [50] and Seo and Shneiderman [88] use zooming in and out to allow the users to focus more into the global structure and find important patterns. Lee et al. [67] enables better readability with a summary of clusters when many nodes in a graph are cluttered and difficult to read. In Xu et al. [113], a TableLens visualization can be used to zoom in on details of individual nodes in the overall clusters in bipartite graphs. Additional visualizations that have been used to support this functionality are scatter plots and dendrograms [31, 88], histograms, bubble charts and chord diagrams [8], box plots [75], matrix-based visualizations [43, 82], and parallel coordinates [31, 98].

7.2.3 One-time Parameter Change. Although many contemporary researchers allow interactions to influence the clustering process, St. Amant and Cohen [95] presents an autonomous, AI-based data exploration assistant (AIDE) that performs clustering and then provides simple results. The user can then query AIDE concerning the choice of parameters and clustering algorithm, but one cannot interact with the clustering process itself. As such, the results are represented but the tool does not iteratively update the clustering process and its results. Some examples found help the user to estimate a suitable number of clusters [64] and to choose different views of the data (e.g., instances of concepts, classes, relations and clusters) through text queries [25]. The user can

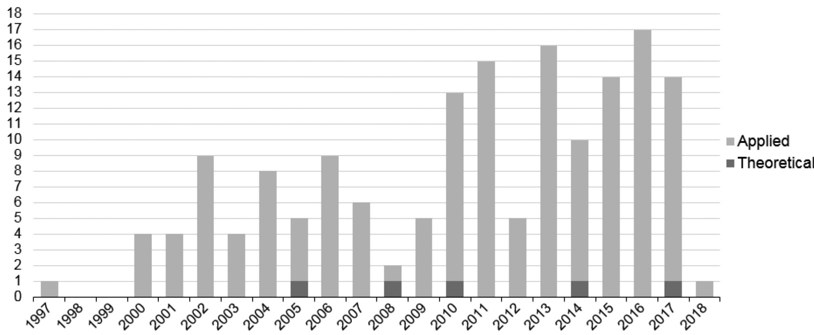

Fig. 2. Distribution of theoretical studies and applied studies over the years.

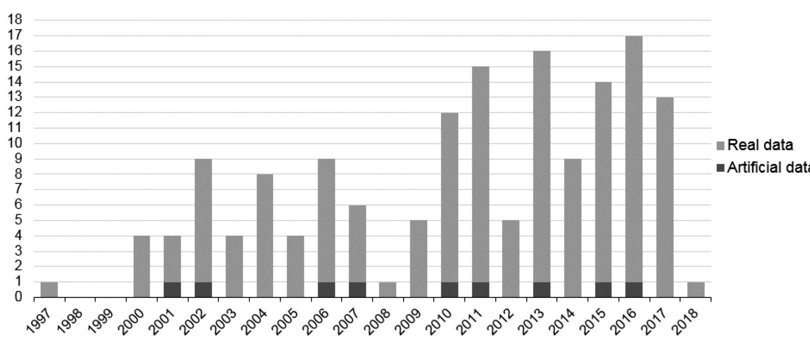

Fig. 3. Distribution of artificial studies and real studies over the years.

filter the results and zoom-in on details (e.g., symptoms or diseases) but not the model itself. Zhou et al. [115] allows zooming out and panning with the location clustering on a map, but clustering results are not updated.

"Direct visual and interactive clustering" among the four conceptual approaches to cluster analysis is supported by the interactive visualization techniques discussed in Hinneburg [52]. This approach consists of replacing automated clustering algorithms (a black box) with interactive, visual, and purely human-operated processes to find clusters. The major challenge therein is finding meaningful visual representations of large volumes of multi- and high-dimensional data. The methods based on scatterplots and parallel coordinate plots are common solutions to this challenge [52]. The examples on parallel coordinates are described in Van Long and Linsen [102] and Dobrev et al. [37].

8 DATASETS

Of the reviewed articles, 100 demonstrate or evaluate clusterings methods on actual datasets, whereas five articles [2, 10, 13, 32, 93] are purely theoretical studies (see Figure 2).

In total, 157 datasets are used, resulting in an average of 1.6 datasets per non-theoretical article. There is a trend of increasing datasets used for non-theoretical article, from a single set per article in 1997 to 2.3 sets per article in 2017. The 157 datasets can be classified into artificial/synthetic data (9 datasets) and real data (148 datasets) (see Figure 3).

In total, 28 (19%) of the real datasets are from general repositories, while the rest (120) are used to solve specific problems/are from identified domains (see Appendix B, Figure 1).

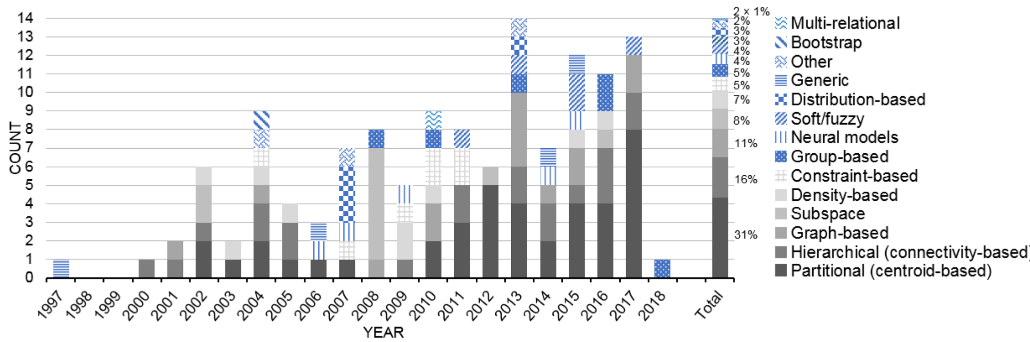

Fig. 4. Distribution of clustering methods per year and over the years. Percentages are in rounded values (e.g., 0.77% as 1, 2.3% as 2).

9 USED CLUSTERING METHODS

In total, 129 clustering methods were reviewed here, which is an average of 1.2 per article (see Appendix B, Table 1 for more details). As can be seen in Figure 4, there is an upward trend of clustering methods used (single method per article in 1997 to 1.6 methods per article in 2017). The clustering methods can be classified into more general categories (types):

Partitional: This category, also referred to as centroid-based clustering, includes the k-means family of methods (k-means, k-medoids, k-medians, fuzzy c-means, etc.) as well as affinity propagation clustering (note that constrained-based k-means clustering is in its own general category).

Hierarchical: This category, also referred to as connectivity-based clustering, includes agglomerative, divisive, and other approaches to hierarchical clustering (e.g., BIRCH).

Graph-based: Includes methods that base clusterings on graph or tree structures. Examples are label propagation algorithms (LPA), non-negative matrix factorization (NNMF), spectral clustering, and other approaches.

Subspace: Includes methods focused on clustering high-dimensional data. Examples are the CLIQUE, DUSC, P3C, PROCLUS, SCHISM, and SUBCLU algorithms.

Density-based: Includes the DBSCAN family of clustering methods and extensions to these methods. Examples are OPTICS and DJ-cluster (density-and-join clustering) algorithms.

Constraint-based: Includes constrained k-means, PCKmeans and other clustering methods that use constraints in the clustering process (e.g., using must-link and must-not-link information).

Group-based: Includes clustering methods that just provide information about the grouping of objects, without a refined clustering model of the results (e.g., methods purely based on split-and-merge operations).

Neural models: Includes clustering methods based on self-organizing maps (SOM).

Soft/fuzzy: Includes methods that are based on the notion of objects belonging to clusters to a certain extent (like the likelihood of being part of a cluster).

Distribution-based: Includes algorithms based on an expectation-maximization (EM) approach. Examples are Gaussian mixture models (GMM), Bayesian mixture models (BMM), multi-modal distributional clustering, and approaches based on probabilistic principal surfaces (PPS).

Generic approaches: These are methods that do not depend on any specific algorithm or class of algorithms.

***Other approaches*:** Includes clustering methods based on bootstrapping techniques, relational data (multi-relational clustering), and tailor-made algorithms that do not fit into other categories.

Figure 4 shows the categorical distribution of clustering methods over time. The largest category is partitional (centroid-based) clustering (40 cases, or 31% of all clustering methods), followed by hierarchical (connectivity-based) (20 cases/16%) and graph-based clustering (14/11%). These three categories constitute more than half of the total number of clustering methods used. Subspace and density-based clustering follow by 10 cases/8% and 9 cases/7%, respectively. Constraint-based clustering, group-based clustering, neural models, and soft/fuzzy clustering form a small group of clustering methods (5–7 cases each, or approximately 5% each). Finally, distribution-based clustering, generic methods, tailor-made ("other") algorithms, bootstrap clustering, and multi-relational clustering are each used one to four times, which together constitutes 10% of the total number of methods.

We find that most partitional (centroid-based) clustering methods used are variations of k-means or specialized versions of k-means (for instance, Babaee et al. [11] uses a modified version called weight-balanced k-means to perform clustering on map images). Kern et al. [63] is the only paper in which a non-parametric method is used (affinity propagation). In hierarchical clustering, all but one method uses an agglomerative approach, with Coden et al. [33] being the only exception by presenting a method based on a divisive strategy. Regarding soft/fuzzy clustering, most cases use topic modeling; Mukhopadhyay et al. [74], being an exception to this, uses a general clustering method based on multi-objective optimization. Sarnovsky [84] introduces an extension to SOM that the authors call "Growing Hierarchical Self-Organizing Map," or GHSOM for short. The GHSOM algorithm dynamically builds a hierarchy of separate SOMs organized into a multi-layered structure. As an example of a tailor-made method, Castellanos-Garzón et al. [26] introduces ClusterBoundary, a novel clustering algorithm capable of finding the objects that constitute the boundaries between clusters.

In Figure 4, it appears that the proportion of graph-based, group-based, and especially, soft/fuzzy clustering methods has increased over the years. Moreover, relative use of partitional (centroid-based) and hierarchical (connectivity-based) clustering methods is quite evenly distributed over the years (see Appendix B, Figure 2).

10 DISCUSSION AND FUTURE WORK

10.1 Overview of the Field

The papers reviewed in this survey cover a broad spectrum of interactivity in the clustering problem. This concerns both the motivation for employing interactivity as well as the technical means of achieving it. The former aspect varies from providing understanding of the results, through subjective evaluation based on individual preferences, all the way to finding a better solution to the optimization problem (essentially claiming superiority of a human over the machine). In the latter aspect, user feedback can be expressed as concrete parameters of the algorithm (e.g., controlling the number of generated clusters) to be incorporated in a very direct manner; or in quite abstract terms (e.g., the request to move a data point from one cluster to another) to be incorporated through complex processing and "interpretation." In fact, many papers fail to distinguish between interactivity embedded in the clustering process, and some used to present clustering results through visualization.

This lack of consensus among papers indicates that there is room for improvement in knowledge and understanding of the field. Very few papers in this area provide comprehensive overview of the state of the art, but instead focus on their particular application area. This makes it difficult to interpret similarities and differences among proposed methods. The only truly universal property is empowering the users to identify cluster-shortcomings. But there is a lack of proper definitions regarding the concept of "interactivity" in this context, i.e., by what criteria is an approach deemed "interactive"?

The primary divergence seems to be related to the underlying motivations. The most common assumption is that a clustering task is too complex to be entirely automated and that user input will lead to "better" (objectively) cluster quality. These solutions can vary in terms of how exactly this quality can be measured and how the interactivity should be implemented, but their common formulation of the problem as an optimization task provides clear structure. On the other end of the spectrum, however, are justifications that focus on the subjective nature of the clustering task. Within this group, there is a much higher variability and disagreement, and the only common ground seems to be realization that no objective "best" clustering exists. Therefore, examples of the goal include better understanding of the data and/or the result, identification of interesting data, or supporting users who approach clustering problems with specific needs. Thus, satisfactory solutions are driven not only by data but also by expectations.

One would expect that these motivations for interactivity would determine how interactivity should be implemented in the sense of defining the operations available to both the user and machine. However, our findings do not confirm this. There is no clear mapping between the goals and the technical means of achieving them. Many papers introduce tools to interact with a model where experts assess the quality of clustering results. In fact, most solutions build upon the same core process: the user sets initial parameters, runs the algorithm, inspects the results through visualizations, (hopefully) obtains new knowledge, makes corrections/refinements, re-runs the algorithm until satisfactory clusters are generated. During the process, the focus is on users and their responsibility to make operations that improve results. This is generally done regardless of "why" (motivation for interactivity) or "when" (which stage the interaction is happening).

10.2 Open Challenges

We have identified this singular scheme to be the most promising area for improvement and future work on the field of interactive clustering. There are a few attempts to create a more comprehensive approach, but only very few papers showcase machine-initiated operations (cf., Section 4.4). There is a need for developing solutions where the machine would initiate quality improving operations, for example, indicate to the users specific clusters that require feedback in a form of a query, and the users would provide information based on such requests. This is a step toward a true mixed-initiative approach where both a user and machine initiate interactions—such a system is yet to be proposed.

One of the main reasons for slow development in the field and the difficulty in comparing competing solutions is that a lot of work is needed today to evaluate state-of-the-art clustering tools. One should consider interactive clustering as a discipline that combines machine learning/data mining (ML/DM) and Human-Computer Interaction (HCI). This is clearly reflected in the evaluations of interactive clustering systems and methods reviewed, where traditional objective performance methods/metrics from ML/DM or subjective assessments from HCI are used. Few papers present a combination of both strategies. It is, however, clear that there is a lack of methods and metrics that can assess overall performance of human–machine collaborations for interactive clustering; that is, ones that go beyond evaluating specific components and consider the system as a whole.

This challenge is definitely noticed, and as the demand for user involvement grows, many researchers mention user study as an area in need of future work [8, 24, 50, 116]. Moreover, Hinneburg [52] addresses a lack of comparative studies and questions how to generally evaluate visual analytics systems. Seo and Shneiderman [89] asks for more case studies that can bring insights and increase understanding of which features are especially effective in certain cases. There has not been, however, any clear progress in this area. One can hope that this will change, however, due to developments in neighbouring disciplines. Works in Explainable AI (XAI) have recently received much attention by both the HCI and AI/ML communities, and we expect that this line of research will also include interactive clustering applications in the near future.

10.3 Synopsis of Future Work

Based on the papers included in this survey we have outlined major challenges identified by the authors, grouped into "technical," "methodology," and "users" categories. This list is clearly far from comprehensive, instead including our subjective selection of more important, relevant, and generic ideas.

Technical improvements. Many studies mention future or ongoing efforts to improve parameter selection (e.g., Sourina and Liu [93], Muller et al. [76], Seo and Shneiderman [89]) and computational efficiency (e.g., Brown et al. [21] to improve re-projection performance with faster eigenvalue calculation methods, or Choo et al. [31], Guo et al. [49] exploring GPU-based clustering). Providing support for selection, tuning, and evaluation metrics for clustering algorithms is also highlighted by several works, such as Seo and Shneiderman [88], Jiang and Canny [62], and El-Assady et al. [42].

Developing new metrics is highlighted as a challenge in Jiang and Canny [62]—particularly for measuring independence of different factors and topic coherence. It is emphasized in several papers to design better visualizations to support exploring and interpreting the clusters (e.g., Chuang and Hsu [32], Hund et al. [60]). For instance, Dudas et al. [41] claims that the difficulty in many domains necessitate combining "user-defined and machine learned clustering," since the final results must match "the user's mental model."

Methodology development. A challenge highlighted by several authors is to compare proposed solutions against other clustering algorithms [7, 31, 49, 50, 64, 69]. For instance, Andrienko et al. [7] points out a need to compare clusters formed from interactive and iterative methods with the results of automated clustering algorithms, but observe that this is currently impossible due to the lack of a scalable clustering algorithm for trajectory data. Liu et al. [69] also mentions to explore such comparisons using other clustering algorithms, especially for investigating relationships between clusters. Rasmussen and Karypis [82] mentions a desire to extend the visualization methods to compare and contrast clustering algorithms over the same datasets.

There is also room for improvement in terms of evaluating methods in different domains and with another data type (e.g., an attempt by Kumpf et al. [64]). In many cases, tools are built for general-purpose use but are only tested in one area (e.g., Rasmussen and Karypis [82] involves a particular expert group and Kwon et al. [65] shows the limitations of health care systems).

Users aspects. Providing better guidance for analysts so they can select the most appropriate clustering algorithm for their task and data is mentioned by Seo and Shneiderman [88], who also identify a need (even for domain experts) for good metrics that distinguish meaningful clusters. Similarly, El-Assady et al. [42] proposes a need for deeper analyses, beyond parameter adjustments, of the black-box of topic models by enabling users to adjust the model in real time as well as providing refinement and optimization suggestions.

Several researchers state that additional user feedback is needed to improve tools and evaluate tool applicability in real-world scenarios (e.g., Boudjeloud-Assala et al. [19], Cao et al. [25], Coden et al. [33], desJardins et al. [35], Hu et al. [55], Lee et al. [67]). Some researchers also list user feedback as vital for developing future tools to improve their performance, for example, by letting

Table 1. Topics of 105 Abstracts of Our Collected Literatures

Topic title	Top 10 terms in each topic
Visual and interaction	*visual interact method analysi subspac set algorithm approach document *dimens
Algorithm and approaches	interact visual algorithm approach result analysi *optim method system *gene
Category and time analysis	visual result analysi interact algorithm group *categor differ *time approach
Feature and document	*featur *document method interact model *improv *constraint framework propos result
Problem and framework	algorithm problem dataset queri model framework set *bound *class interact
Interface and queries	*interfac *queri model result approach *search interact present integr system

Top 10 terms indicate globally common terms of the topic. The words with * (asterisk) indicate the "unique terms focused" on that specific topic with relevance value 0.2 (within top 10). All terms in the table were revealed with relevance 1, and terms with * are with relevance 0.2. Refer to Reference [91].

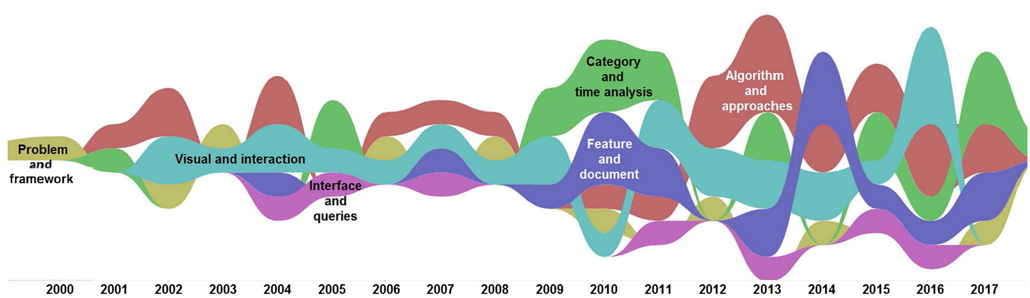

Fig. 5. Topic flow of 105 abstracts on interactive clustering. *Visual and interaction* and *algorithm and approaches* are dominant topics.

users evaluate various constraints together with possible newer constraints [36]. Others argue that users should be part of the quality assurance process by incorporating user feedback into tool assessments (e.g., Bruneau and Otjacques [22]). Another open question is how to effectively integrate user feedback at varying abstraction and generalization levels, especially when feedback is coupled with cluster restructuring [54]. To this end, Dubey et al. [40] discusses several challenges related to improving user feedback.

In many domain-specific applications, human expertise can significantly improve the clustering results. For example, Porter et al. [80] states that "user bias can help steer simple classifiers to better solutions, and in fact, bias is critical to interactive machine learning reaching its full potential." However, more research is needed to formally distinguish the positive effects of human (or user) bias (i.e., domain expertise) from its negative effects (i.e., human factors) in terms of interactive clustering.

10.4 Topic Modeling on Collected Literatures (Using Latent Dirichlet Allocation)

We performed topic modeling with latent Dirichlet allocation [12, 18, 91] and found six main topics (Table 1). Figure 5 illustrates the topic trend of collected 105 paper's abstracts from the year 2000. *Visual and interaction* is the dominant topic, followed by *algorithm & approaches*, *category & time analysis*, *feature & document*, *problem & framework*, and *interface & queries*. We included stopwords such as cluster(s), user, use, and data that appeared in most of the topics. The topic titles are given by the authors based on the top 10 terms from each topic.

Overall, we reviewed 105 papers chronicling different clustering methods, which included their various purposes, types of interactive operations, and data sources. Moreover, we discovered many evaluation methods along with efforts to iteratively incorporate user feedback for better understanding. Yet, there remains no general tool to resolve the challenges mentioned, and users still need to fully understand iterative and interactive clustering processes and results.

A SUPPLEMENTARY MATERIALS

See the supplementary materials in the online version.

REFERENCES

[1] Zafar Ahmed and Chris Weaver. 2012. An adaptive parameter space-filling algorithm for highly interactive cluster exploration. In *Proceedings of the 2012 IEEE Conference on Visual Analytics Science and Technology (VAST'12)*. 13–22.

[2] Nir Ailon, Anup Bhattacharya, Ragesh Jaiswal, and Amit Kumar. 2018. Approximate clustering with same-cluster queries. In *9th Innovations in Theoretical Computer Science Conference (ITCS'18) and Leibniz International Proceedings in Informatics (LIPIcs)*, Anna R. Karlin (Ed.), Vol. 94. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 40:1–40:21. https://doi.org/10.4230/LIPIcs.ITCS.2018.40

[3] Saleema Amershi, Maya Cakmak, William Bradley Knox, and Todd Kulesza. 2014. Power to the people: The role of humans in interactive machine learning. *AI Mag.* 35, 4 (2014), 105–120.

[4] Gennady Andrienko and Natalia Andrienko. 2010. Interactive cluster analysis of diverse types of spatiotemporal data. *ACM SIGKDD Explor. Newslett.* 11, 2 (May 2010), 19–28.

[5] Gennady Andrienko and Natalia Andrienko. 2015. Visualization support to interactive cluster analysis. In *Machine Learning and Knowledge Discovery in Databases*. Springer International Publishing, 337–340.

[6] Gennady Andrienko, Natalia Andrienko, Salvatore Rinzivillo, Mirco Nanni, and Dino Pedreschi. 2009. A visual analytics toolkit for cluster-based classification of mobility data. In *Advances in Spatial and Temporal Databases*, Nikos Mamoulis, Thomas Seidl, Torben Bach Pedersen, Kristian Torp, and Ira Assent (Eds.), Lecture Notes in Computer Science, Vol. 5644. Springer, Berlin, 432–435.

[7] Gennady Andrienko, Natalia Andrienko, Salvatore Rinzivillo, Mirco Nanni, Dino Pedreschi, and Fosca Giannotti. 2009. Interactive visual clustering of large collections of trajectories. In *Proceedings of the 2009 IEEE Symposium on Visual Analytics Science and Technology*. 3–10.

[8] İnanç Arın, Mert Kemal Erpam, and Yücel Saygın. 2018. I-TWEC: Interactive clustering tool for Twitter. *Expert Syst. Appl.* 96 (2018), 1–13.

[9] Pranjal Awasthi, Maria-Florina Balcan, and Konstantin Voevodski. 2017. Local algorithms for interactive clustering. *Journal of Machine Learning Research* 18, 3 (2017), 1–35. http://jmlr.org/papers/v18/15-085.html.

[10] Pranjal Awasthi and Reza B. Zadeh. 2010. Supervised clustering. In *Advances in Neural Information Processing Systems*, J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylora, R. S. Zemel, and A. Culotta (Eds.). Curran Associates, Inc., 91–99.

[11] Mohammadreza Babaee, Reza Bahmanyar, Gerhard Rigoll, and Mihai Datcu. 2014. Interactive clustering for SAR image understanding. In *Proceedings of the 10th European Conference on Synthetic Aperture Radar (EUSAR'14)*. 1–4.

[12] Juhee Bae, Jesper Havsol, Martin Karpefors, Alexander Karlsson, and Gunnar Mathiason. 2019. Short text topic modeling to identify trends on wearable bio-sensors in different media types. In *Proceedings of the 6th International Symposium on Computational and Business Intelligence*.

[13] Maria-Florina Balcan and Avrim Blum. 2008. Clustering with Interactive Feedback. In *Algorithmic Learning Theory*, Yoav Freund, László Györfi, György Turán, and Thomas Zeugmann (Eds.), Lecture Notes in Computer Science, Vol. 5254. Springer, Berlin, Heidelberg, 316–328.

[14] Sugato Basu, Arindam Banerjee, and Raymond J. Mooney. 2004. Active semi-supervision for pairwise constrained clustering. In *Proceedings of the 2004 SIAM International Conference on Data Mining*, Michael W. Berry, Umeshwar Dayal, Chandrika Kamath, and David Skillicorn (Eds.). Society for Industrial and Applied Mathematics, Philadelphia, PA, 333–344.

[15] Sumit Basu, Danyel Fisher, Steven M. Drucker, and Hao Lu. 2010. Assisting users with clustering tasks by combining metric learning and classification. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI'10)*. American Association for Artificial Intelligence, 394–400.

[16] Ron Bekkerman, Hema Raghavan, James Allan, and Koji Eguchi. 2007. Interactive clustering of text collections according to a user-specified criterion. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'07)*. 684–689.

[17] Michael R. Berthold, Bernd Wiswedel, and David E. Patterson. 2002. Neighborgram clustering. Interactive exploration of cluster neighborhoods. In *Proceedings of the 2002 IEEE International Conference on Data Mining*. 581–584.

[18] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet allocation. *J. Mach. Learn. Res.* 3 (Mar. 2003), 993–1022.

[19] Lydia Boudjeloud-Assala, Philippe Pinheiro, Alexandre Blansché, Thomas Tamisier, and Benoît Otjacques. 2016. Interactive and iterative visual clustering. *Inf. Vis.* 15, 3 (2016), 181–197.

[20] Joel Brandt, Jiayi Chong, and Sean Rosenbaum. 2006. Interactive clustering for data exploration.

[21] Eli T. Brown, Jingjing Liu, Carla E. Brodley, and Remco Chang. 2012. Dis-function: Learning distance functions interactively. In *Proceedings of the 2012 IEEE Conference on Visual Analytics Science and Technology (VAST'12).* 83–92.

[22] Pierrick Bruneau and Benoît Otjacques. 2013. An interactive, example-based, visual clustering system. In *Proceedings of the 2013 17th International Conference on Information Visualisation.* 168–173.

[23] Pierrick Bruneau, Philippe Pinheiro, Bertjan Broeksema, and Benoît Otjacques. 2015. Cluster Sculptor, an interactive visual clustering system. *Neurocomputing* 150, Part B (2015), 627–644.

[24] Nan Cao, David Gotz, Jimeng Sun, and Huamin Qu. 2011. DICON: Interactive visual analysis of multidimensional clusters. *IEEE Trans. Vis. Comput. Graph.* 17, 12 (Dec. 2011), 2581–2590.

[25] Nan Cao, Jimeng Sun, Yu-Ru Lin, David Gotz, Shixia Liu, and Huamin Qu. 2010. FacetAtlas: Multifaceted visualization for rich text corpora. *IEEE Trans. Vis. Comput. Graph.* 16, 6 (Nov. 2010), 1172–1181.

[26] José A. Castellanos-Garzón, Carlos Armando García, Paulo Novais, and Fernando Díaz. 2013. A visual analytics framework for cluster analysis of DNA microarray data. *Expert Syst. Appl.* 40, 2 (2013), 758–774.

[27] Shuo Chang, Peng Dai, Lichan Hong, Cheng Sheng, Tianjiao Zhang, and Ed H. Chi. 2016. AppGrouper: Knowledge-based interactive clustering tool for app search results. In *Proceedings of the 21st International Conference on Intelligent User Interfaces (IUI'16).* ACM, New York, NY, 348–358.

[28] Keke Chen and Ling Liu. 2003. Validating and refining clusters via visual rendering. In *Proceedings of the 3rd IEEE International Conference on Data Mining.* 501–504.

[29] Keke Chen and Ling Liu. 2006. iVIBRATE: Interactive visualization-based framework for clustering large datasets. *ACM Trans. Inf. Syst.* 24, 2 (Apr. 2006), 245–294.

[30] Jaegul Choo, Changhyun Lee, Chandan K. Reddy, and Haesun Park. 2013. UTOPIAN: User-driven topic modeling based on interactive nonnegative matrix factorization. *IEEE Trans. Vis. Comput. Graph.* 19, 12 (Dec. 2013), 1992–2001.

[31] Jaegul Choo, Hanseung Lee, Zhicheng Liu, John Stasko, and Haesun Park. 2013. An interactive visual testbed system for dimension reduction and clustering of large-scale high-dimensional data. In *Proceedings of the Visualization and Data Analysis 2013*, Vol. 8654. International Society for Optics and Photonics.

[32] Jason Chuang and Daniel J. Hsu. 2014. Human-centered interactive clustering for data analysis. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS'14) Workshop on Human-Propelled Machine Learning.*

[33] Anni Coden, Marina Danilevsky, Daniel Gruhl, Linda Kato, and Meena Nagarajan. 2017. A method to accelerate human in the loop clustering. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, Nitesh Chawla and Wei Wang (Eds.). Society for Industrial and Applied Mathematics, 237–245.

[34] Geraldo N. Corrêa, Ricardo M. Marcacini, Eduardo R. Hruschka, and Solange O. Rezende. 2015. Interactive textual feature selection for consensus clustering. *Pattern Recogn. Lett.* 52 (2015), 25–31.

[35] Marie desJardins, James MacGlashan, and Julia Ferraioli. 2007. Interactive visual clustering. In *Proceedings of the 12th International Conference on Intelligent User Interfaces (IUI'07).* ACM, New York, NY, 361–364.

[36] Hugo Alexandre Dantas do Nascimento and Peter Eades. 2001. Interactive graph clustering based on user hints. In *Proceedings of the 2nd International Workshop on Soft Computing Applied to Software Engineering (SCASE'01)*, Jens H. Jahnke and Conor Ryan (Eds.). 99–105.

[37] Petar Dobrev, Tran Van Long, and Lars Linsen. 2011. A cluster hierarchy-based volume rendering approach for interactive visual exploration of multi-variate volume data. In *Vision, Modeling, and Visualization*, Peter Eisert, Joachim Hornegger, and Konrad Polthier (Eds.). The Eurographics Association.

[38] Vladimir Dobrynin, David Patterson, Mykola Galushka, and Niall Rooney. 2005. SOPHIA: An interactive cluster-based retrieval system for the OHSUMED collection. *IEEE Trans. Inf. Technol. Biomed.* 9, 2 (Jun. 2005), 256–265.

[39] Fei Du, A-Xing Zhu, and Feng Qi. 2016. Interactive visual cluster detection in large geospatial datasets based on dynamic density volume visualization. *Geocarto Int.* 31, 6 (2016), 597–611.

[40] Avinava Dubey, Indrajit Bhattacharya, and Shantanu Godbole. 2010. A cluster-level semi-supervision model for interactive clustering. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD'10)*, José Luis Balcázar, Francesco Bonchi, Aristides Gionis, and Michèle Sebag (Eds.). Lecture Notes in Computer Science, Vol. 6321. Springer, Berlin, 409–424.

[41] Patrick M. Dudas, Martijn De Jongh, and Peter Brusilovsky. 2013. A semi-supervised approach to visualizing and manipulating overlapping communities. In *Proceedings of the 17th International Conference on Information Visualisation*, Vol. 1. 180–185.

[42] Mennatallah El-Assady, Rita Sevastjanova, Fabian Sperrle, Daniel Keim, and Christopher Collins. 2017. Progressive learning of topic modeling parameters: A visual analytics framework. *IEEE Trans. Vis. Comput. Graph.* 24, 1 (2017), 382–391. https://doi.org/10.1109/TVCG.2017.2745080

[43] Ugo Erra, Bernardino Frola, and Vittorio Scarano. 2011. An interactive bio-inspired approach to clustering and visualizing datasets. In *Proceedings of the 15th International Conference on Information Visualisation*. 440–447.

[44] Adil Fahad, Najlaa Alshatri, Zahir Tari, Abdullah Alamri, Ibrahim Khalil, Albert Y. Zomaya, Sebti Foufou, and Abdelaziz Bouras. 2014. A survey of clustering algorithms for big data: Taxonomy and empirical analysis. *IEEE Trans. Emerg. Top. Comput.* 2, 3 (Sep. 2014), 267–279.

[45] Benoit Gaudin and Aaron J. Quigley. 2008. Interactive structural clustering of graphs based on multi-representations. In *Proceedings of the 12th International Conference Information Visualisation*. 227–232.

[46] Floris Geerts and Reuben Ndindi. 2014. Interactive correlation clustering. In *Proceedings of the 2014 International Conference on Data Science and Advanced Analytics (DSAA'14)*. 170–176.

[47] Alicja Gruźdź, Aleksandra Ihnatowicz, and Dominik Ślęzak. 2006. Interactive gene clustering—A case study of breast cancer microarray data. *Inf. Syst. Front.* 8, 1 (1 Feb. 2006), 21–27.

[48] Diansheng Guo, Donna J. Peuquet, and Mark Gahegan. 2003. ICEAGE: Interactive clustering and exploration of large and high-dimensional geodata. *GeoInformatica* 7, 3 (1 Sep. 2003), 229–253.

[49] Peihong Guo, He Xiao, Zuchao Wang, and Xiaoru Yuan. 2010. Interactive local clustering operations for high dimensional data in parallel coordinates. In *Proceedings of the 2010 IEEE Pacific Visualization Symposium (PacificVis'10)*. 97–104.

[50] Steffen Hadlak, Heidrun Schumann, Clemens H. Cap, and Till Wollenberg. 2013. Supporting the visual analysis of dynamic networks by clustering associated temporal attributes. *IEEE Trans. Vis. Comput. Graph.* 19, 12 (Dec. 2013), 2267–2276.

[51] Anne-Wil Harzing. 2007. Publish or Perish. Retrieved from https://harzing.com/resources/publish-or-perish.

[52] Alexander Hinneburg. 2014. Concepts of visual and interactive clustering. In *Data Clustering: Algorithms and Applications*, Charu C. Aggarwal and Chandan K. Reddy (Eds.). CRC Press, 483–504.

[53] Enamul Hoque and Giuseppe Carenini. 2015. ConVisIT: Interactive topic modeling for exploring asynchronous online conversations. In *Proceedings of the 20th International Conference on Intelligent User Interfaces (IUI'15)*. ACM, New York, NY, 169–180.

[54] M. Shahriar Hossain, Praveen Kumar Reddy Ojili, Cindy Grimm, Rolf Müller, Layne T. Watson, and Naren Ramakrishnan. 2012. Scatter/gather clustering: Flexibly incorporating user feedback to steer clustering results. *IEEE Trans. Vis. Comput. Graph.* 18, 12 (Dec. 2012), 2829–2838.

[55] Yuening Hu, Jordan Boyd-Graber, and Brianna Satinoff. 2011. Interactive topic modeling. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (HLT'11)*. Association for Computational Linguistics, Stroudsburg, PA, 248–257.

[56] Yeming Hu, Evangelos E. Milios, and James Blustein. 2010. *Interactive Document Clustering Using Iterative Class-based Feature Selection*. Technical Report CS-2010-04. Faculty of Computer Science, Dalhousie University, Halifax, Nova Scotia, Canada.

[57] Yeming Hu, Evangelos E. Milios, and James Blustein. 2011. Interactive feature selection for document clustering. In *Proceedings of the 2011 ACM Symposium on Applied Computing (SAC'11)*. ACM, New York, NY, 1143–1150.

[58] Yeming Hu, Evangelos E. Milios, and James Blustein. 2014. Interactive document clustering with feature supervision through reweighting. *Intell. Data Anal.* 18, 4 (2014), 561–581.

[59] Zhexue Huang, Michael K. Ng, Tao Lin, and David Cheung. 2000. An interactive approach to building classification models by clustering and cluster validation. In *Proceedings of the 2nd International Conference on Intelligent Data Engineering and Automated Learning (IDEAL'00). Data Mining, Financial Engineering, and Intelligent Agents*, Kwong Sak Leung, Lai-Wan Chan, and Helen Meng (Eds.), Lecture Notes in Computer Science, Vol. 1983. Springer, Berlin, 23–28.

[60] Michael Hund, Dominic Böhm, Werner Sturm, Michael Sedlmair, Tobias Schreck, Torsten Ullrich, Daniel A. Keim, Ljiljana Majnaric, and Andreas Holzinger. 2016. Visual analytics for concept exploration in subspaces of patient groups. *Brain Inf.* 3, 4 (1 Dec. 2016), 233–247.

[61] Francesco Iorio, Gennaro Miele, Francesco Napolitano, Giancarlo Raiconi, and Roberto Tagliaferri. 2007. An interactive tool for data visualization and clustering. In *Proceedings of the Knowledge-Based Intelligent Information and Engineering Systems (KES'07)*, Bruno Apolloni, Robert J. Howlett, and Lakhmi Jain (Eds.). Lecture Notes in Computer Science, Vol. 4694. Springer-Verlag, Berlin, 870–877.

[62] Biye Jiang and John F. Canny. 2015. Interactive clustering with a high-performance ML toolkit. In *Proceedings of the KDD 2015 Workshop on Interactive Data Exploration and Analytics (IDEA'15)*. 37–46.

[63] Michael Kern, Alexander Lex, Nils Gehlenborg, and Chris R. Johnson. 2017. Interactive visual exploration and refinement of cluster assignments. *BMC Bioinf.* 18, 1, Article 406 (12 Sep. 2017), 406.

[64] Alexander Kumpf, Bianca Tost, Marlene Baumgart, Michael Riemer, Rüdiger Westermann, and Marc Rautenhaus. 2018. Visualizing confidence in cluster-based ensemble weather forecast analyses. *IEEE Trans. Vis. Comput. Graph.* 24, 1 (Jan 2018), 109–119.

[65] Bum Chul Kwon, Ben Eysenbach, Janu Verma, Kenney Ng, Christopher De Filippi, Walter F. Stewart, and Adam Perer. 2017. Clustervision: Visual supervision of unsupervised clustering. *IEEE Trans. Vis. Comput. Graph.* 24, 1 (29 Aug. 2017), 142–151.

[66] Hien Phuong Lai, Muriel Visani, Alain Boucher, and Jean-Marc Ogier. 2014. A new interactive semi-supervised clustering model for large image database indexing. *Pattern Recogn. Lett.* 37 (2014), 94–106.

[67] Hanseung Lee, Jaeyeon Kihm, Jaegul Choo, John Stasko, and Haesun Park. 2012. iVisClustering: An interactive visual document clustering via topic modeling. In *Computer Graphics Forum*, Vol. 31. 1155–1164.

[68] Yang Lei, Dai Yu, Zhang Bin, and Yang Yang. 2017. Interactive K-means clustering method based on user behavior for different analysis target in medicine. *Computational and Mathematical Methods in Medicine* 2017 (July 2017), 9.

[69] Wei Liu, Gui-Rong Xue, Shen Huang, and Yong Yu. 2005. Interactive Chinese search results clustering for personalization. In *Proceedings of the 6th International Conference on Advances in Web-Age Information Management (WAIM'05))()*, Wenfei Fan, Zhaohui Wu, and Jun Yang (Eds.), Lecture Notes in Computer Science, Vol. 3735. Springer, Berlin, 676–681.

[70] Carl G. Looney. 2002. Interactive clustering and merging with a new fuzzy expected value. *Pattern Recogn.* 35, 11 (2002), 2413–2423.

[71] Sehi L'Yi, Bongkyung Ko, DongHwa Shin, Young-Joon Cho, Jaeyong Lee, Bohyoung Kim, and Jinwook Seo. 2015. XCluSim: A visual analytics tool for interactively comparing multiple clustering results of bioinformatics data. *BMC Bioinf.* 16, 11 (2015), S5.

[72] Joseph MacInnes, Stephanie Santosa, and William Wright. 2010. Visual classification: Expert knowledge guides machine learning. *IEEE Comput. Graph. Appl.* 30, 1 (Jan. 2010), 8–14.

[73] Logan Adam Mitchell. 2016. *INCREMENT—Interactive Cluster Refinement.* Master's thesis. Brigham Young University, Provo, UT.

[74] Anirban Mukhopadhyay, Ujjwal Maulik, and Sanghamitra Bandyopadhyay. 2013. An interactive approach to multiobjective clustering of gene expression patterns. *IEEE Trans. Biomed. Eng.* 60, 1 (Jan. 2013), 35–41.

[75] Emmanuel Müller, Ira Assent, Ralph Krieger, Timm Jansen, and Thomas Seidl. 2008. Morpheus: Interactive exploration of subspace clustering. In *Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* ACM, New York, NY, 1089–1092.

[76] Wolfgang Muller, Soufyane El Allali, Daniel Blank, Andreas Henrich, and Thomas Lauterbach. 2007. Hunt the cluster: A scalable, interactive time bayesian image browser for P2P networks. In *Proceedings of the 9th IEEE International Symposium on Multimedia Workshops (ISMW'07).* IEEE.

[77] Seyednaser Nourashrafeddin, Evangelos Milios, and Dirk Arnold. 2013. Interactive text document clustering using feature labeling. In *Proceedings of the 2013 ACM Symposium on Document Engineering (DocEng'13).* ACM, New York, NY, 61–70.

[78] Masayuki Okabe and Seiji Yamada. 2011. An interactive tool for human active learning in constrained clustering. *J. Emerg. Technol. Web Intell.* 3, 1 (2011), 20–27.

[79] Eli Packer, Peter Bak, Mikko Nikkilä, Valentin Polishchuk, and Harold J. Ship. 2013. Visual analytics for spatial clustering: Using a heuristic approach for guided exploration. *IEEE Trans. Vis. Comput. Graph.* 19, 12 (Dec. 2013), 2179–2188.

[80] Reid Porter, James Theiler, and Don Hush. 2013. Interactive machine learning in data exploitation. *Comput. Sci. Eng.* 15, 5 (2013), 12–20.

[81] Teng Qiu and Yongjie Li. 2015. IT-map: An effective nonlinear dimensionality reduction method for interactive clustering. *ArXiv e-prints* (2015).

[82] Matt Rasmussen and George Karypis. 2004. *gCLUTO—An Interactive Clustering, Visualization, and Analysis System.* Technical Report. Department of Computer Science and Engineering, University of Minnesota.

[83] Tim Rawlins, Andrew Lewis, Jan Hettenhausen, and Seyedali Mirjalili. 2012. Interactive k-means clustering for investigation of optimisation solution data. In *Proceedings of the 16th Biennial Computational Techniques and Applications Conference (CTAC'12)*, Ian Turner (Ed.). Australian Mathematical Society, 1–2.

[84] Martin Sarnovsky. 2014. Design and implementation of Interactive visualization of GHSOM clustering algorithm for text mining tasks. *Int. J. Res. Inf. Technol.* 1, 7 (Jul. 2014), 146–151.

[85] Yusuke Sato and Makoto Iwayama. 2009. Interactive constrained clustering for patent document set. In *Proceedings of the 2nd International Workshop on Patent Information Retrieval (PaIR'09).* ACM, New York, NY, 17–20.

[86] Tobias Schreck, Jürgen Bernard, Tatiana Von Landesberger, and Jörn Kohlhammer. 2009. Visual cluster analysis of trajectory data with interactive kohonen maps. *Inf. Vis.* 8, 1 (Jan. 2009), 14–29.

[87] Jessica Zeitz Self, Michelle Dowling, John Wenskovitch, Ian Crandell, Ming Wang, Leanna House, Scotland Leman, and Chris North. 2018. Observation-level and parametric interaction for high-dimensional data analysis. *ACM Trans. Interact. Intell. Syst.* 8, 2 (Jun. 2018), 15:1–15:36. DOI: https://doi.org/10.1145/3158230

[88] Jinwook Seo and Ben Shneiderman. 2002. Interactively exploring hierarchical clustering results. *Computer* 35, 7 (Jul. 2002), 80–86.

[89] Jinwook Seo and Ben Shneiderman. 2005. *Using Categorical Information in Multidimensional Data Sets: Interactive Partition and Cluster Comparison.* Technical Report TR 2005-102. Institute for Systems Research, University of Maryland, College Park, MD.

[90] Ben Shneiderman. 1996. The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings of the IEEE Symposium on Visual Languages.* 336–343.

[91] Carson Sievert and Kenneth Shirley. 2014. LDAvis: A method for visualizing and interpreting topics. In *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces.* Association for Computational Linguistics, Baltimore, MD, 63–70.

[92] Harri Siirtola. 2004. Interactive cluster analysis. In *Proceedings of the 8th International Conference on Information Visualisation (IV'04).* 471–476.

[93] Olga Sourina and Dongquan Liu. 2005. Visual interactive clustering and querying of spatio-temporal data. In *Proceedings of the 2005 International Conference on Computational Science and Its Applications (ICCSA'05)*, Vol. 4. Springer-Verlag, Berlin, 968–977.

[94] Akash Srivastava, James Zou, and Charles Sutton. 2016. Clustering with a reject option: Interactive clustering as Bayesian prior elicitation. *ArXiv e-prints* (7 2016), 16–20.

[95] Robert St. Amant and Paul R. Cohen. 1997. Interaction with a mixed-initiative system for exploratory data analysis. In *Proceedings of the 2nd International Conference on Intelligent User Interfaces (IUI'97).* ACM, New York, NY, 15–22.

[96] Rodger Staden. 1984. Measurements of the effects that coding for a protein has on a DNA sequence and their use for finding genes. *Nucleic Acids Res.* 12, 1(2) (1984), 555–567.

[97] Yasufumi Takama and Takuma Tonegawa. 2016. Interactive document clustering system based on coordinated multiple views. *J. Adv. Comput. Intell. Intell. Inf.* 20, 1 (2016), 139–145.

[98] Andrada Tatu, Fabian Maaß, Ines Färber, Enrico Bertini, Tobias Schreck, Thomas Seidl, and Daniel Keim. 2012. Subspace search and visualization to make sense of alternative clusterings in high-dimensional data. In *Proceedings of the 2012 IEEE Conference on Visual Analytics Science and Technology (VAST'12).* 63–72.

[99] Cagatay Turkay, Alexander Lex, Marc Streit, Hanspeter Pfister, and Helwig Hauser. 2014. Characterizing cancer subtypes using dual analysis in Caleydo StratomeX. *IEEE Comput. Graph. Appl.* 34, 2 (Mar. 2014), 38–47.

[100] Cagatay Turkay, Július Parulek, Nathalie Reuter, and Helwig Hauser. 2011. Interactive visual analysis of temporal cluster structures. *Comput. Graph. Forum* 30, 3 (2011), 711–720.

[101] Cagatay Turkay, Július Parulek, Nathalie Reuter, and Helwig Hauser. 2013. Integrating cluster formation and cluster evaluation in interactive visual analysis. In *Proceedings of the 27th Spring Conference on Computer Graphics (SCCG'11).* ACM, New York, NY, 77–86.

[102] Tran Van Long and Lars Linsen. 2009. MultiClusterTree: Interactive visual exploration of hierarchical clusters in multidimensional multivariate data. *Comput. Graph. Forum* 28, 3 (2009), 823–830.

[103] Sharad Vikram and Sanjoy Dasgupta. 2016. Interactive Bayesian hierarchical clustering. In *Proceedings of the 33rd International Conference on Machine Learning*, Maria Florina Balcan and Kilian Q. Weinberger (Eds.), Proceedings of Machine Learning Research, Vol. 48. New York, NY.

[104] Ulrike Von Luxburg. 2010. Clustering stability: An overview. *Found. Trends Mach. Learn.* 2, 3 (2010), 235–274.

[105] XiuFeng Wan, Susan M. Bridges, and John A. Boyle. 2004. Revealing gene transcription and translation initiation patterns in archaea, using an interactive clustering model. *Extremophiles* 8, 4 (1 Aug. 2004), 291–299.

[106] Xiufeng Wan, Susan M. Bridges, John A. Boyle, and Alan P. Boyle. 2002. Interactive clustering for exploration of genomic data. *SmartEng Des.* 12 (July 2002), 753–758.

[107] Xiang Wang and Ian Davidson. 2010. Active spectral clustering. In *Proceedings of the 2010 IEEE International Conference on Data Mining.* 561–568.

[108] Zhangye Wang, Chang Chen, Juanxia Zhou, Jiyuan Liao, Wei Chen, and Ross Maciejewski. 2013. A novel visual analytics approach for clustering large-scale social data. In *Proceedings of the 2013 IEEE International Conference on Big Data.* 79–86.

[109] John Wenskovitch, Ian Crandell, Naren Ramakrishnan, Leanna House, and Chris North. 2018. Towards a systematic combination of dimension reduction and clustering in visual analytics. *IEEE Trans. Vis. Comput. Graph.* 24, 1 (Jan. 2018), 131–141.

[110] Wensheng Wu, Clement Yu, AnHai Doan, and Weiyi Meng. 2004. An interactive clustering-based approach to integrating source query interfaces on the deep web. In *Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data (SIGMOD'04).* ACM, New York, NY, 95–106.

[111] Yongqiao Xiao and Margaret H. Dunham. 2001. Interactive clustering for transaction data. In *Data Warehousing and Knowledge Discovery*, Yahiko Kambayashi, Werner Winiwarter, and Masatoshi Arikawa (Eds.), Lecture Notes in Computer Science, Vol. 2114. Springer, Berlin, 121–130.

[112] Caiming Xiong, David M. Johnson, and Jason J. Corso. 2017. Active clustering with model-based uncertainty reduction. *IEEE Trans. Pattern Anal. Mach. Intell.* 39, 1 (Jan. 2017), 5–17.

[113] Panpan Xu, Nan Cao, Huamin Qu, and John Stasko. 2016. Interactive visual co-cluster analysis of bipartite graphs. In *Proceedings of the 2016 IEEE Pacific Visualization Symposium (PacificVis'16)*. 32–39.

[114] Li Zhang, Chun Tang, Yong Shi, Yuqing Song, Aidong Zhang, and Murali Ramanathan. 2002. VizCluster: An interactive visualization approach to cluster analysis and its application on microarray data. In *Proceedings of the 2002 SIAM International Conference on Data Mining*. SIAM, Philadelphia, PA, 19–40.

[115] Changqing Zhou, Dan Frankowski, Pamela Ludford, Shashi Shekhar, and Loren Terveen. 2004. Discovering personal gazetteers: An interactive clustering approach. In *Proceedings of the 12th Annual ACM International Workshop on Geographic Information Systems (GIS'04)*. ACM, New York, NY, 266–273.

[116] Fangfang Zhou, Juncai Li, Wei Huang, Ying Zhao, Xiaoru Yuan, Xing Liang, and Yang Shi. 2016. Dimension reconstruction for visual exploration of subspace clusters in high-dimensional data. In *Proceedings of the 2016 IEEE Pacific Visualization Symposium (PacificVis'16)*. 128–135.

[117] Lina Zhou, Shimei Pan, Jianwu Wang, and Athanasios V. Vasilakos. 2017. Machine learning on big data: Opportunities and challenges. *Neurocomputing* 237 (May 2017), 350–361.

Received September 2018; revised June 2019; accepted June 2019