The Cognitive and Computational Benefits and Limitations of Clustering for Sensemaking

John Wenskovitch
Michelle Dowling
Chris North
Discovery Analytics Center
Virginia Tech
Blacksburg, VA, USA
jw87@cs.vt.edu
dowlingm@cs.vt.edu
north@cs.vt.edu

Abstract

The cognitive process of sensemaking refers to acquiring, representing, and organizing information in order to understand that information. The organization component naturally supports the introduction of clusters, an important enabler for grouping objects such that similar objects are placed in the same cluster. This paper explores the benefits and limitations of introducing clusters into systems for exploratory data analysis. We consider these issues for tasks that the system may support, methods for visualizing and interacting with data in the system, and algorithms that are encoded into the system. We discuss the use of clusters in these systems with respect to cognition and computation, and we call out future areas of research in this area.

Author Keywords

Sensemaking; clustering; exploratory data analysis; tasks; visualization; interaction.

ACM Classification Keywords

H.5.m [Information interfaces and presentation (e.g., HCI)]: Miscellaneous

Introduction

Sensemaking refers to a cognitive process for acquiring, representing, and organizing information in order to address a task, solve a problem, or make a decision [31, 22].

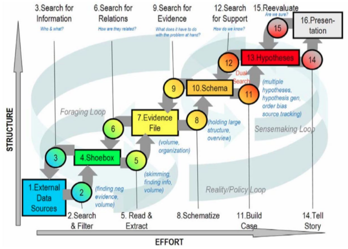


Figure 1: In the Sensemaking Process, intelligence analysts transform raw information into reportable results through organizational stages that filter, extract, and structure pieces of data. There are several natural leverage points for clustering in this process.

A number of models with varying levels of information granularity have been proposed for approaching and solving sensemaking problems [30, 31, 29]. These models represent strategies for addressing a variety of sensemaking problems. For example, Pirolli and Card's Sensemaking Process [30] is designed for sensemaking problems faced by intelligence analysts. Despite the specific challenge addressed by each of these models, they all highlight the need to organize the data. For example, an intelligence analyst may work to understand the actors and motivations by grouping documents by location, by person, or by subplot.

A fundamental behavior in sensemaking is the act of grouping similar observations[1] in order to understand their properties, effectively forming a cluster. This organizational strategy is true both in paper-based sensemaking tasks [14, 42] and in tasks performed on electronic displays [4, 15]. Clusters therefore have a natural connection to sensemaking. Clusters can also help to reduce clutter in a workspace, compressing similar observations into a group that requires less physical or screen space [27]. Simplifying the workspace leads to further cognitive benefits, as humans struggle to think about more than a small number of observations or dimensions at one time [33]. Thus, using groups of items to perform analysis tasks can lead to improved memory and recall by providing a simplified method of understanding the data [12].

However, clusters are inherently subjective structures, making the identification of clusters by humans a challenging process that is often problem-specific. Previous research has shown that humans use a variety of organizational principles to cluster information [13], even when addressing the same task [4]. In order to identify clusters computationally, hundreds of clustering algorithms have been implemented, each with strengths and weaknesses. As a result, there is no universally optimal clustering algorithm. Instead, the best clustering algorithm to solve a problem is often determined experimentally [17].

Our contribution in this work is an overview of the cognitive and computational benefits and limitations of clustering for sensemaking in exploratory data analysis. Rather than providing a complete survey of this field, we discuss the features of several representative visualization techniques designed to explore datasets. We begin with a discussion of common sensemaking tasks that can be supported by clusters in these techniques. Given these tasks, we describe visualizations and interactions that can be implemented on those clusters, and conclude with a discussion of clustering algorithms that support these visualizations and interactions for sensemaking. In these sections, we discuss both the cognitive benefits and limitations of clustering, as well as the computational benefits and limitations of clustering. We also raise a number of research questions that can be addressed as exploration into this design space continues.

Tasks

In this section, we discuss the different types of tasks that clustering supports using the list of clustering tasks defined by Wenskovitch et al. [40], and how they connect to the low-level analysis tasks by Amar et al. [3]. These tasks directly support different steps in the Sensemaking Process by Pirolli and Card [30], and therefore reflect more cognitive benefits of clustering rather than computational. However, a benefit that is reflected both cognitively and computationally is scalability. By clustering observations, an analyst can

[1] In this work, we employ the convention of referring to the features (columns) of a dataset as *dimensions*, individual data items (rows) as *observations*, and the features of those observations (cells) as *attributes*. *Node* is used to indicate the glyph representing an observation in a visualization.

Task Research Questions:

1. Are there any tasks that are inherently hindered by the inclusion of clustering, regardless of implementation or visualization?
2. Are there any tasks that are supported exclusively by clustering, that cannot be addressed by any other algorithm, visualization, or interaction?

perform sensemaking tasks with bigger datasets while still retaining the ability to interpret the visualization.

To begin our evaluation of the exploratory data analysis tasks supported by clustering, we first turn to the list of representative clustering tasks listed by Wenskovitch et al. [40]. We use these to analyze which tasks are most or least commonly supported amongst exploratory techniques that leverage clustering of high-dimensional data. For example, identifying clusters, seeing their relative positions, and determining cluster structure are clearly the most common tasks supported [32, 23, 2, 18, 11, 41, 21, 8, 25], as they should be for tools such as these which are designed to provide insights into clusters of a dataset. In comparison, support for tasks to explore the data are more varied, with changing cluster membership of observations being the most common task of this type supported [23, 11, 41]. The next most commonly supported task in this category is creating or removing clusters [23, 11]. Although direct support for repositioning clusters in the projection space isn't common [11], this is more often indirectly supported through other tasks, such as manipulating a parameter of the clustering algorithm [23, 41]. Understanding the data via the tasks of labeling clusters [18, 28, 21, 8] is a less commonly supported task.

In comparing the clustering tasks defined by Wenskovitch et al. [40] with the low-level analysis tasks defined by Amar et al. [3], we note that these tasks overlap. Of course, there is the obvious, broad overlap in that one of Amar's analysis tasks is clustering itself. For exploratory data analysis techniques, clustering also typically implies computing derived values for each cluster (e.g., most salient dimension or topic within each cluster) [23, 10, 21, 8]. Given that finding a high or low extrema attribute within a given cluster is directly supported by these computed derived values, clustering techniques that compute derived values generally support this analysis task as well. Similarly, techniques that show cluster structure directly support characterizing the distribution within that structure [32, 23, 2, 18, 11, 41, 21, 8, 25]. This information can help to uncover correlations between clusters or observations as well as to find anomalies. Other tasks such as retrieving values are commonly supported [23, 2, 10, 41, 8], which may additionally help determine ranges [10]. Filtering [10] and sorting [23, 10, 28] are occasionally supported analysis tasks in exploratory data technique that utilize clustering.

Beyond these general categories of tasks that are supported by clustering, some exploratory data analysis techniques are designed to support specific tasks. Many of these techniques highlight support for refining the clustering results themselves [23, 40, 21, 8], which is a task that is supported mathematically through a combination of visualization and clustering techniques. As another example, iVisClustering [23], Termite [10], ClustVis [28], and UTOPIAN [8] all afford tasks such as understanding which dimensions or terms best describe a given cluster. This task is dimension-centric, as opposed to understanding cluster structure, which is an observation-centric task. Additionally, the technique described by Wenskovitch and North [41] supports the task of determining which dimensions describe the differences between the clusters at a global level.

All of the tasks supported by these exploratory data analysis techniques demonstrate how clusters can be leveraged to improve sensemaking. That is, these tasks directly support sensemaking tasks. For example, in Pirolli and Card's Sensemaking Process, clustering in general supports organizing of tasks (which helps the analyst in the "Evidence File" step) or skimming of the data (proceeding from the

"Shoebox" step to the "Evidence File" step). Similarly, clustering provides an overview of the data and imposes structure, which helps the analyst proceed from the "Evidence File" step to the "Schema" step. Filtering and sorting may assist with these two steps of the sensemaking loop in addition to enabling searching for specific types of data (proceeding from the "External Data Sources" step to the "Shoebox" step). Thus, any additional task supported by an exploratory data analysis technique that leverages clustering, including those described previously, only further enhances its ability to support sensemaking.

Visualization Research Questions:

1. If the number of clusters in a dataset can vary widely, how can a visualization be designed to naturally transition from one cluster membership representation to another when the original encoding is no longer effective?
2. What preattentive features (aside from color and position) can be used to visualize clusters efficiently in future applications?
3. After clustering has been introduced, is there a limit to the size of a dataset that can be visualized?

Through this section, we described the cognitive benefits of clustering for sensemaking, but an open question remains: **Are there any tasks that are inherently hindered by the inclusion of clustering, regardless of implementation or visualization?** It is certainly true that specific implementations of clustering visualizations can prevent tasks such as identifying extrema and determining range from being completed, but other visualizations can still support these tasks. Are there any cases where a task is universally harmed by the inclusion of a clustering algorithm? In a similar vein, **are there any tasks that are supported exclusively by clustering, that cannot be addressed by any other algorithm, visualization, or interaction?** In other words, is clustering required to accomplish some tasks, or is there always another way to accomplish the task?

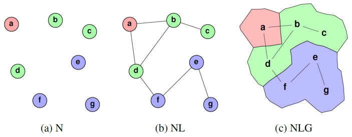
Visualization

The variety of methods for visualizing clusters is nearly as broad as the variety of clustering algorithms. Among others, these techniques include encoding cluster membership in color, in position, and in distinctly-separated groups. Such visual encodings assist with tasks such as identifying clusters. In cases where position is used, tasks such as seeing relative positions of clusters, determining cluster structure, finding correlations, and finding anomalies can also be supported. Here, we discuss several example systems that encode cluster membership using these three techniques.

Using color to indicate cluster membership has been demonstrated in a number of tools and prototypes. Saket et al. [32] demonstrated three different methodologies for using color in this manner: coloring nodes, coloring nodes in node-link diagrams, and coloring regions of node-link diagrams. The iVisClustering tool [23] demonstrates this first methodology of using color. Linesets [2] uses colored nodes in node-link diagrams as well as colored links when connecting nodes with the same cluster membership. Lastly, coloring regions of node-link diagrams can be found in GMap [18], which renders a geographic-like map for clusters. Bubble Sets [11] shows an interesting mixture of the Linesets and GMap visualizations by drawing isocontours around nodes and links to form clusters.

Leveraging color encodings to indicate cluster membership affords preattentive recognition of these clusters [20]. However, there are also cognitive limits to using color to encode cluster membership. For example, the human eye has trouble distinguishing between more than 10 distinct colors in a visualization [34]. As such, color encodings are better suited for visualizations with a small number of clusters. A natural follow-up research question is **if the number of clusters in a dataset can vary widely, how can a visualization be designed to naturally transition from one cluster membership representation to another when the original encoding is no longer effective?**

Position can also be used to encode cluster information using a variety of methods. The Termite system [10] visualizes the relationship between words and topics in a Term-Topic Matrix, a 2D matrix indexed by computed topics on one axis and words on the other. When sorted using their seriation



(a) N (b) NL (c) NLG

Figure 2: Saket et al. evaluate three options for encoding cluster membership, relating each to the effectiveness of performing node- and group-based tasks.

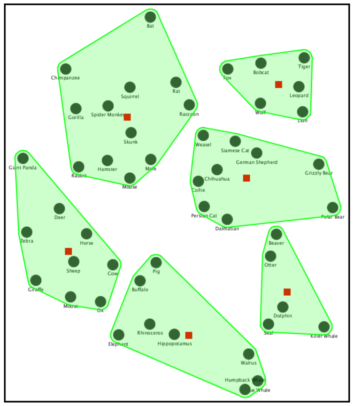


Figure 3: Wenskovitch and North use the k-means algorithm on a dimension-reduced projection to cluster observations.

technique, cluster of terms appear as vertical stripes of larger nodes across the words in the matrix. A related technique is seen in the heatmap view of ClustVis [28], which displays dendrograms along the heatmap axes to display the hierarchical cluster structure. The rows and columns of the heatmap are thus positioned based on the ordering of the clustering tree. Cognitively, position is not a preattentive feature, though it can be used to complement closure (discussed next). However, node size is a preattentive feature [20], which is also demonstrated by Termite [10].

A third method for visualizing cluster information is via distinct separation boundaries (i.e., closure), which is another preattentive feature that can be exploited [20] for fast cognitive recognition of clusters. For example, Wenskovitch and North [41] work to combine dimension reduction and clustering algorithms in the same projection space. In this projection, cluster memberships are encoded by drawing convex hulls around each cluster (see Figure 3). Because k-means is limited to finding convex clusters, a convex hull is a natural visual representation for the output of this algorithm. The scatterplot view from ClustVis [28] contains clear boundaries between clusters by drawing ellipses encompassing the observations categorized within each cluster, and the space-filling group encoding studied by Saket et al. [32] has similar, clearly-delineated regions.

A visualization is not limited to choosing only one technique; dual-encoding [19] is a frequently used technique to reinforce cluster membership. For example, both TopicLens [21] and UTOPIAN [8] use position (via modified versions of t-SNE [25]) and color to encode clusters of documents. Since dual-encoding aids in visualizing clusters, another open research questions is, **What other preattentive features can be used to visualize clusters efficiently in future applications?**

In addition to cognitive benefits, there are computational benefits to clustering in visualizations as well, particularly as the datasets grow large. For example, consider a visualization system that is given millions of observations to visualize. Visualizing all observations will be computationally slow and cognitively overwhelming. Clustering observations and only visualizing the clusters at some level in the hierarchy reduces the workload on both the computer and the visual system of the analyst. An implementation of this strategy can be found in ASK-GraphView [1]. Given that clustering data into a hierarchy has been shown to benefit scalability, an open research question is: **After clustering has been introduced, is there a limit to the size of a dataset that can be visualized (assuming that hardware can support that dataset)?**

However, a major trade-off in only visualizing the clusters is that tasks such as determining cluster structure [40], determining range, or characterizing distribution [3] are no longer readily supported through the visualization. Thus, creating such a visualization imposes a cognitive limitation for the analyst in addition to the aforementioned cognitive benefit.

Another limitation is that visualizations must have a default initial display of the observations and clusters. Presenting an analyst with this initial projection could produce a cognitive limitation by biasing their exploration towards patterns or structure that they notice. This could lead to a further difficulty if the initial projection is misleading by creating cluster memberships that bias the analyst's investigation towards a similar solution instead of enabling any solution. Some techniques for exploratory data analysis recognize these drawbacks, attempting to compensate for them by using randomized initial displays or by providing interaction methods to learn how to cluster the entire dataset based on a few user-driven observation classifications [41].

Interaction Research Questions:

1. How can systems for exploratory data analysis learn to interpret analyst interactions as updates to clustering parameters?
2. What types of interactions do analysts naturally want to perform on clusters?

Interaction

Simply presenting an analyst with a visualization may be sufficient for the analyst to perform tasks such as identify clusters, seeing relative positions of clusters, determining cluster structure, find anomalies, or find correlations. However, Malone shows that categorizing information is an important factor in organization to improve the cognition of data [26]. Additionally, sensemaking naturally requires exploring the data beyond the initial projection to draw further insight from updated visualizations. For example, tasks such as retrieving a value or computing a derived value is typically not information displayed for every cluster when a dataset is first visualized. Other tasks such as repositioning clusters, changing cluster membership, and creating or moving clusters are not supported at all. Thus, interaction is a necessary component of exploring clusters during data analysis. Although a number of taxonomies and studies have been generated for interacting with high-dimensional data [7, 38, 44], our focus in this sections is primarily on interaction strategies and mechanisms found in the visualizations that we discussed in the previous section.

To begin, most previously mentioned exploratory data analysis techniques afford an interaction to provide the analyst with more information about an observation or about a cluster in order to better understand the layout and grouping of the observations. These interactions are almost universally details-on-demand via mouseover [23, 41, 21, 8], though there are other methods that support the acquisition of contextual information. For example, Termite [10] allows analysts to click on a term to view its distribution across the entire dataset, as well as to click on a topic to view its representative documents. TopicLens [21] provides a resizable mouseover lens that dynamically divides the overlaid subset of observations within the lens into subclusters, enabling an analyst to see finer-grained structure among the observations. Analysts are also able to filter contextual information to only the most salient observations [10].

Analysts can also supply contextual information for their own use as they explore the data, such as labeling clusters [23]. Contextual information is not limited to the labels and contents of observations and clusters. Some systems supply cluster similarity information [23], as well as sorting [23, 10, 28] and coloring [28] mechanisms to support other tasks such as characterizing distributions. No change is made to clustering assignments or observation layout with any of these interactions; instead, these interactions are simply providing the analyst with contextual information about the observations and clusters.

Many systems also provide analysts with standard GUI widgets such as dropdown menus, slider bars, and checkboxes to alter layout and clustering parameters in the visualization. In ClustVis [28], these parametric interactions allow an analyst to alter the clustering method, linkage method, and the sorting order of the rows and columns in the matrix independently. ClustVis also allows an analyst to change which principal components are used for the axes in its scatterplot view. UTOPIAN [8] uses these controls in a sidebar to enable an analyst to modify both the dimension reduction and clustering algorithm parameters, as well as in a popup to alter the term importances that define the clusters. iVisClustering [23] also provides parameter sliders to manipulate the cluster algorithm parameters, allowing the analyst to directly adjust cluster assignments. These interactions afford cognitive benefits by enabling tasks such as changing cluster membership of observations. However, these benefits rely on the analyst's ability to understand the clustering algorithm used and how the interactions affect the projection of the data. For example, some algorithms like t-SNE [25], are difficult for novice analysts to understand due to the intricacies of the algorithm and its parameters [39]. Thus, only analysts who understand the clustering algorithm receive these cognitive benefits. Because efficient and correct use of a clustering algorithm also depends on understand its behavior, these analysts miss out computational efficiency and benefits as well.

Some exploratory data analysis techniques also give analysts the ability to directly manipulate the observations and clusters, thereby affording additional tasks. For example, observations can be selected and dragged to repositioning them within clusters or to relocate them to other clusters [11, 41]. Clusters can also be selected and moved [11], joined and split [8, 23], created [8, 41, 23], removed [2, 23], and increased in prominence or importance [2]. A cognitive advantage of these interactions is that algorithms can be created which will re-cluster all data based on how the analyst changes cluster memberships. Effectively, the system will tune clustering parameters on behalf of the analyst.

All of these interaction techniques are centered around one research question: **How can systems for exploratory data analysis learn to interpret user interactions as updates to clustering parameters?** In each of the interactions described, the analyst directly manipulates either the visualization (e.g., labeling clusters) or some parameter for the clustering algorithm (e.g., manipulating a slider). However, not all of the aforementioned exploratory data analysis techniques use these interactions to alter cluster parameters; only certain interactions, including the aforementioned repositioning of observations within the visualization, are used in this manner, which provide intuitive yet powerful methods for exploring the data. Thus, a natural supporting research question is, **What types of interactions do analysts naturally want to perform on clusters?** The answer to this question indicates what interactions should be afforded and, based on the analyst's expectations are, what the results of the interactions should be (i.e., how the interaction should be interpreted).

Algorithms

Behind each of these visualizations and interactions lies one or more clustering algorithms to organize and structure the data, both initially as well as based on user interactions. Other works have surveyed the design space of clustering algorithms [9, 43]. We focus on a few representative examples, both common clustering algorithms and those that were used in tools that we discussed in previous sections.

Perhaps the most commonly-used clustering algorithm is k-means, which partitions a dataset into k clusters according to a distance between each data item and the nearest cluster centroid. Cognitively, this implies that k-means will identify clusters based on nearest neighbors via some distance measure and that clusters will consist of very similar observations. Finding an optimal solution via k-means is an NP-hard problem, resulting in the creation of algorithms that yield heuristic clustering solutions. The running time of Lloyd's algorithm, for example, is O($nkdi$) for a dataset with n items of d dimensions each, k clusters, and i iterations before convergence [24]. Thus, this algorithm is linear in terms of both the size of the dataset and the intended number of clusters. Though efficient, the fundamental limitation is that it can only detect convex clusters, and often will converge to a non-optimal solution when noise is present.

Density-based clustering algorithms such as DBSCAN [16] and OPTICS [5] overcome this issue by seeking out clusters in areas that have higher density. By doing so, sparse regions clearly identify and separate clusters, and the occasional observations between are treated as noise. Cognitively, these algorithms locate dense regions of obser-

Algorithm Research Questions:

1. Which clustering algorithms should be chosen to support the desired interaction and visualization techniques?
2. How do we make the parameters of the clustering algorithms understandable to the analyst?
3. How should we design an interface to enable understandable parameter tuning?
4. How should clustering algorithms learn from complex interactions?

vations, leading to an interpretation that the number of observations in a region is more important than individual similarity relationships located by k-means. Because density-based algorithms can learn the number of clusters dynamically, no k parameter for number of clusters sought is necessary; however, these algorithms require a distance threshold to determine the location of the cluster boundary.

Dirichlet process clustering [36, 35] has gained popularity in the recent past with statisticians. Rather than returning a hard clustering assignment, this is a probabilistic method that computes a probability that an observation under consideration belongs to each cluster. Cognitively, this algorithm replaces the hard clustering assignment of the previously-discussed algorithms. Like density-based clustering, Dirichlet Process clustering learns the number of clusters dynamically, but it does not require a distance threshold. It does, however, require the specification of a probability distribution for the observations in each cluster. Despite these computational advantages, the runtime of this algorithm scales poorly as the number of observations increases, a clear computational challenge.

Dimension reduction algorithms are used for embedding high-dimensional data into a 2D (or occasionally 3D) projection such that similarities in the structure of the high-dimensional data such as clusters and outliers are preserved in the low-dimensional projection. Cognitively, this aids an analyst in thinking in a more familiar spatial layout than an incomprehensible n-dimensional space, but the projection could result in the introduction of false neighbors (which could propagate into false clustering assignments) as information is naturally lost when reducing from n dimensions to two dimensions. One such dimension reduction algorithm is t-distributed Stochastic Neighbor Embedding (t-SNE) [25], which constructs probability distributions in the high-dimensional and low-dimensional spaces and minimizes distances between them. Both TopicLens [21] and UTOPIAN [8] use a modified version of the original t-SNE algorithm to project their clusters into 2D visualizations.

Latent Dirichlet Allocation (LDA) [6] is a topic modeling algorithm, an area of research that works to discover abstract “topics” that are present in a collection of documents based on keywords contained in those documents. Each topic can therefore be thought of as a cluster, which contains documents most relevant to that topic. The iVisClustering tool [23] uses LDA for topic modeling, and hence for document clustering.

One computational advantage of clustering is the improvement it provides on the runtime of other algorithms. For example, many dimension reduction algorithms have runtimes of O(n^2) or even O(n^3) for n observations to project. Grouping observations into clusters and only visualizing the clusters can easily greatly reduce the runtime of the dimension reduction algorithms. For example, grouping these n observations into clusters with an average size of 10 presents a $\sim$100x performance boost for O(n^2) dimension reduction algorithms or $\sim$1000x for O(n^3) algorithms.

A major clustering limitation is that many algorithms rely on one or more input parameters, such as the set number of clusters in k-means, the distance threshold in density-based clustering, and the probability distribution in Dirichlet process clustering. This means that, at the very least, the initial projection must choose a default for these parameters. Attempting to dynamically determine these parameters is possible but also computationally expensive, which will cause delays in displaying the visualization. Using k-means again as an example, a frequently used technique is to run the algorithm repeatedly with various values of k, and then selecting the optimal value based on diminishing returns in the reduction of variance as k increases (otherwise known as the "elbow method" [37]). Of course, running a clustering algorithm m times to learn a parameter rather than once increases the computational complexity by a factor of m.

In these automated methods to computationally identify these parameters, the analyst's knowledge is not used. This can lead to a cognitive drawback in which the analyst may disagree with initial clustering of the data. This issue can be alleviated by allowing analysts to change cluster membership of observations. Taking this a step further, the system can capture such interactions and learn how to cluster other observations in the dataset, reducing the number of interactions needed to produce a desirable clustering of the data. As discussed previously, such interactivity can be accomplished at an algorithmic level by having the system tune clustering parameters on behalf of the analyst.

In addition to these trade-offs of cognitive and computational effects, these algorithms are used to support certain types of interactions, such as directly manipulating clustering algorithm parameters (e.g., the sliders in UTOPIAN [8]), or visualizations, such as ensuring clusters are represented by non-overlapping convex hulls (e.g., the visualization technique by Wenskovitch and North. [41]. For example, t-SNE clusters observations, but may result in all observations being pulled into clusters depending on the parameter values used. This may result in a visualization that obscures some characteristics of the data such as outliers, thereby hindering tasks such as seeing the relative positions of clusters or finding anomalies. Thus, the chosen clustering algorithm must be compatible with the desired visualization and interaction techniques to support the chosen tasks. This leads to another research question: **Which clustering algorithm should be chosen to support the desired interaction and visualization techniques?**

Depending on the desired interaction and visualization techniques, there may be many follow-up research questions. For example, in a visualization in which the analyst can directly manipulate clustering parameters, **how can we make the parameters of the clustering algorithm understandable to the analyst?** In other words, how can the parameters of the clustering algorithm be represented in such a way that the analyst understands how each parameter influences the clustering algorithm? While this may be a simple question to answer when considering expert analysts, it is more complicated for novice analysts who may know little about the chosen clustering algorithm. Similarly, **how should we design a visualization to enable understandable parameter tuning?** The analyst may want to perform a specific interaction (e.g., manipulate a slider) to change a parameter value. This natural interaction method should be supported by the visualization and result in appropriate changes to the clustering algorithm parameters.

For more complex interactions, the system should translate the interaction to manipulations of model parameters on behalf of the analyst. Thus, an open research question is, **how should clustering algorithms learn from complex interactions?** When the analyst changes the cluster assignment of an observation, what can the system infer from this interaction? What parameters of the algorithm should be manipulated to reflect this change across all clusters? How should these parameters be manipulated?

Conclusion

Through our discussion of these techniques, we described many of the cognitive and computational benefits of using clustering for sensemaking. Clustering and sensemaking is a natural pairing, as clustering provides the ability to group observations and interact with those groups: a necessity for the organizational component of sensemaking.

To briefly summarize, we saw cognitive benefits of clustering amongst tasks, visualization methods, interaction methods, and algorithms. These benefits include scalability, ease of understanding patterns in the data, updating the visualization to reflect user intent, and communicating the details of the clustering to the analyst. We also saw computational benefits of clustering for tasks, visualization methods, and algorithms. These benefits include efficient rendering, less cluttered visualizations, and improved interactivity. Additionally, we noted some limitations of clustering, including the need for parameter tuning and understandability and potentially misleading default clustering assignments. Finally, we proposed a number of future research questions in each section, ranging from alternative methods of handling clustering tasks to selecting a clustering algorithm.

We note one strong limitation of the analysis presented in this work. We have not fully surveyed or taxonomized this space to provide an exhaustive list of benefits and drawbacks of clustering, nor have we cited an exhaustive list of papers that use clustering for sensemaking. Instead, we chose representative examples of clustering used in exploratory data analysis techniques combined with research that we have undertaken. This may have biased some of our statements and analysis throughout the preceding sections towards what we judged to be most important cognitive and computational benefits of clustering in exploratory data analysis. In the future, this survey of benefits and limitations in exploratory data analysis techniques may be expanded into a more thorough survey.

Acknowledgements

This research was supported by NSF Grant IIS-1447416. The authors would also like to thank fellow research group members for interesting discussions that led to some ideas presented in this work.

REFERENCES

1. James Abello, Frank van Ham, and Neeraj Krishnan. 2006. ASK-GraphView: A Large Scale Graph Visualization System. *IEEE Transactions on Visualization and Computer Graphics* 12, 5 (Sept. 2006), 669–676. DOI: http://dx.doi.org/10.1109/TVCG.2006.120

2. B. Alper, N. Riche, G. Ramos, and M. Czerwinski. 2011. Design Study of LineSets, a Novel Set Visualization Technique. *IEEE Transactions on Visualization and Computer Graphics* 17, 12 (Dec 2011), 2259–2267. DOI: http://dx.doi.org/10.1109/TVCG.2011.186

3. R. Amar, J. Eagan, and J. Stasko. 2005. Low-level components of analytic activity in information visualization. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.* 111–117. DOI: http://dx.doi.org/10.1109/INFVIS.2005.1532136

4. Christopher Andrews, Alex Endert, and Chris North. 2010. Space to Think: Large High-resolution Displays for Sensemaking. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. ACM, New York, NY, USA, 55–64. DOI: http://dx.doi.org/10.1145/1753326.1753336

5. Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, and Jörg Sander. 1999. OPTICS: Ordering Points to Identify the Clustering Structure. In *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data (SIGMOD '99)*. ACM, New York, NY, USA, 49–60. DOI: http://dx.doi.org/10.1145/304182.304187

6. David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.

7. Andreas Buja, Dianne Cook, and Deborah F. Swayne. 1996. Interactive High-Dimensional Data Visualization. *Journal of Computational and Graphical Statistics* 5, 1 (1996), 78–99. DOI: http://dx.doi.org/10.1080/10618600.1996.10474696

8. J. Choo, C. Lee, C. K. Reddy, and H. Park. 2013. UTOPIAN: User-Driven Topic Modeling Based on Interactive Nonnegative Matrix Factorization. *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (Dec 2013), 1992–2001. DOI: http://dx.doi.org/10.1109/TVCG.2013.212

9. Jason Chuang and Daniel J Hsu. 2014. Human-Centered Interactive Clustering for Data Analysis. *Conference on Neural Information Processing Systems (NIPS). Workshop on Human-Propelled Machine Learning* (2014).

10. Jason Chuang, Christopher D. Manning, and Jeffrey Heer. 2012. Termite: Visualization Techniques for Assessing Textual Topic Models. In *Proceedings of the International Working Conference on Advanced Visual Interfaces (AVI '12)*. ACM, New York, NY, USA, 74–77. DOI: http://dx.doi.org/10.1145/2254556.2254572

11. C. Collins, G. Penn, and S. Carpendale. 2009. Bubble Sets: Revealing Set Relations with Isocontours over Existing Visualizations. *IEEE Transactions on Visualization and Computer Graphics* 15, 6 (Nov 2009), 1009–1016. DOI: http://dx.doi.org/10.1109/TVCG.2009.122

12. Jacqueline M Curiel and Gabriel A Radvansky. 1998. Mental organization of maps. *Journal of Experimental Psychology: Learning, Memory, and Cognition* 24, 1 (1998), 202.

13. Paul Dourish, John Lamping, and Tom Rodden. 1999. Building Bridges: Customisation and Mutual Intelligibility in Shared Category Management. In *Proceedings of the International ACM SIGGROUP Conference on Supporting Group Work (GROUP '99)*. ACM, New York, NY, USA, 11–20. DOI: http://dx.doi.org/10.1145/320297.320299

14. Steven M. Drucker, Danyel Fisher, and Sumit Basu. 2011. Helping Users Sort Faster with Adaptive Machine Learning Recommendations. In *Human-Computer Interaction – INTERACT 2011*, Pedro Campos, Nicholas Graham, Joaquim Jorge, Nuno Nunes, Philippe Palanque, and Marco Winckler (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 187–203.

15. Alex Endert, Seth Fox, Dipayan Maiti, Scotland Leman, and Chris North. 2012. The Semantics of Clustering: Analysis of User-generated Spatializations of Text Documents. In *Proceedings of the International Working Conference on Advanced Visual Interfaces (AVI '12)*. ACM, New York, NY, USA, 555–562. DOI: http://dx.doi.org/10.1145/2254556.2254660

16. Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, and others. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise.. In *Kdd*, Vol. 96. 226–231.

17. Vladimir Estivill-Castro. 2002. Why So Many Clustering Algorithms: A Position Paper. *SIGKDD Explor. Newsl.* 4, 1 (June 2002), 65–75. DOI: http://dx.doi.org/10.1145/568574.568575

18. E. R. Gansner, Y. Hu, and S. Kobourov. 2010. GMap: Visualizing graphs and clusters as maps. In *2010 IEEE Pacific Visualization Symposium (PacificVis)*. 201–208. DOI:http://dx.doi.org/10.1109/PACIFICVIS.2010.5429590

19. Rex Hartson and Pardha S Pyla. 2012. *The UX Book: Process and guidelines for ensuring a quality user experience*. Elsevier.

20. Christopher G. Healey, Kellogg S. Booth, and James T. Enns. 1996. High-speed Visual Estimation Using Preattentive Processing. *ACM Trans. Comput.-Hum. Interact.* 3, 2 (June 1996), 107–135. DOI: http://dx.doi.org/10.1145/230562.230563

21. M. Kim, K. Kang, D. Park, J. Choo, and N. Elmqvist. 2017. TopicLens: Efficient Multi-Level Visual Topic Exploration of Large-Scale Document Collections. *IEEE Transactions on Visualization and Computer Graphics* 23, 1 (Jan 2017), 151–160. DOI: http://dx.doi.org/10.1109/TVCG.2016.2598445

22. Gary Klein, Brian Moon, and Robert R Hoffman. 2006. Making sense of sensemaking 2: A macrocognitive model. *IEEE Intelligent systems* 21, 5 (2006), 88–92.

23. Hanseung Lee, Jaeyeon Kihm, Jaegul Choo, John Stasko, and Haesun Park. 2012. iVisClustering: An Interactive Visual Document Clustering via Topic Modeling. *Computer Graphics Forum* 31, 3pt3 (2012), 1155–1164. DOI:http://dx.doi.org/10.1111/j.1467-8659.2012.03108.x

24. S. Lloyd. 1982. Least squares quantization in PCM. *IEEE Transactions on Information Theory* 28, 2 (Mar 1982), 129–137. DOI: http://dx.doi.org/10.1109/TIT.1982.1056489

25. Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, Nov (2008), 2579–2605.

26. Thomas W. Malone. 1983. How Do People Organize Their Desks?: Implications for the Design of Office Information Systems. *ACM Trans. Inf. Syst.* 1, 1 (Jan. 1983), 99–112. DOI: http://dx.doi.org/10.1145/357423.357430

27. Richard Mander, Gitta Salomon, and Yin Yin Wong. 1992. A &Ldquo;Pile&Rdquo; Metaphor for Supporting Casual Organization of Information. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '92)*. ACM, New York, NY, USA, 627–634. DOI: http://dx.doi.org/10.1145/142750.143055

28. Tauno Metsalu and Jaak Vilo. 2015. ClustVis: a web tool for visualizing clustering of multivariate data using Principal Component Analysis and heatmap. *Nucleic acids research* 43, W1 (2015), W566–W570.

29. Peter Pirolli and Stuart Card. 1999. Information foraging. *Psychological review* 106, 4 (1999), 643.

30. Peter Pirolli and Stuart Card. 2005. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. *Proceedings of International Conference on Intelligence Analysis* 5 (2005), 2–4.

31. Daniel M. Russell, Mark J. Stefik, Peter Pirolli, and Stuart K. Card. 1993. The Cost Structure of Sensemaking. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems (CHI '93)*. ACM, New York, NY, USA, 269–276. DOI: http://dx.doi.org/10.1145/169059.169209

32. B. Saket, P. Simonetto, S. Kobourov, and K. BÃűrner. 2014. Node, Node-Link, and Node-Link-Group Diagrams: An Evaluation. *IEEE Transactions on Visualization and Computer Graphics* 20, 12 (Dec 2014), 2231–2240. DOI: http://dx.doi.org/10.1109/TVCG.2014.2346422

33. 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 Transactions on Interactive Intelligent Systems* ***[IN PRESS]*** (2018).

34. Robert Simmon. Use of Color in Data Visualization. (????). https://earthobservatory.nasa.gov/resources/blogs/intro_to_color_for_visualization.pdf

35. Yee Whye Teh. 2011. Dirichlet process. In *Encyclopedia of machine learning*. Springer, 280–287.

36. Yee Whye Teh, Michael I Jordan, Matthew J Beal, and David M Blei. 2005. Sharing clusters among related groups: Hierarchical Dirichlet processes. In *Advances in neural information processing systems*. 1385–1392.

37. Robert L. Thorndike. 1953. Who belongs in the family? *Psychometrika* 18, 4 (01 Dec 1953), 267–276. DOI: http://dx.doi.org/10.1007/BF02289263

38. Tatiana von Landesberger, Sebastian Fiebig, Sebastian Bremm, Arjan Kuijper, and Dieter W. Fellner. 2014. *Interaction Taxonomy for Tracking of User Actions in Visual Analytics Applications*. Springer New York, New York, NY, 653–670. DOI: http://dx.doi.org/10.1007/978-1-4614-7485-2_26

39. Martin Wattenberg, Fernanda ViÃľgas, and Ian Johnson. 2016. How to Use t-SNE Effectively. *Distill* (2016). DOI: http://dx.doi.org/10.23915/distill.00002

40. John Wenskovitch, Ian Crandell, Naren Ramakrishnan, Leanna House, Scotland Leman, and Chris North. 2018. Towards a Systematic Combination of Dimension Reduction and Clustering in Visual Analytics. *IEEE Transactions on Visualization and Computer Graphics* 24, 1 (Jan 2018), 131–141. DOI: http://dx.doi.org/10.1109/TVCG.2017.2745258

41. John Wenskovitch and Chris North. 2017. Observation-Level Interaction with Clustering and Dimension Reduction Algorithms. In *Proceedings of the 2nd Workshop on Human-In-the-Loop Data Analytics (HILDA'17)*. ACM, New York, NY, USA, Article 14, 6 pages. DOI: http://dx.doi.org/10.1145/3077257.3077259

42. Steve Whittaker and Julia Hirschberg. 2001. The Character, Value, and Management of Personal Paper Archives. *ACM Trans. Comput.-Hum. Interact.* 8, 2 (June 2001), 150–170. DOI: http://dx.doi.org/10.1145/376929.376932

43. Rui Xu and D. Wunsch. 2005. Survey of clustering algorithms. *IEEE Transactions on Neural Networks* 16, 3 (May 2005), 645–678. DOI: http://dx.doi.org/10.1109/TNN.2005.845141

44. J. S. Yi, Y. a. Kang, and J. Stasko. 2007. Toward a Deeper Understanding of the Role of Interaction in Information Visualization. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (Nov 2007), 1224–1231. DOI: http://dx.doi.org/10.1109/TVCG.2007.70515