

Observation-Level Interaction with Clustering and Dimension Reduction Algorithms

John Wenskovitch, Chris North
Discovery Analytics Center, Virginia Tech
Blacksburg, VA
{ jw87,north } @vt.edu

ABSTRACT

Observation-Level Interaction (OLI) is a sensemaking technique relying upon the interactive semantic exploration of data. By manipulating data items within a visualization, users provide feedback to an underlying mathematical model that projects multidimensional data into a meaningful two-dimensional representation. In this work, we propose, implement, and evaluate an OLI model which explicitly defines clusters within this data projection. These clusters provide targets against which data values can be manipulated. The result is a cooperative framework in which the layout of the data affects the clusters, while user-driven interactions with the clusters affect the layout of the data points. Additionally, this model addresses the OLI “with respect to what” problem by providing a clear set of clusters against which interaction targets are judged and computed.

CCS Concepts

•Human-centered computing → Visualization techniques;

Keywords

Observation-Level Interaction (OLI), sensemaking, data clustering, semantic interaction, visual analytics

1. INTRODUCTION

Observation-Level Interaction (OLI) is a direct manipulation technique that supports sensemaking by allowing users to actively manipulate the layout of data within a visualization. Through these interactions, an underlying model is updated, effectively letting the visualization system learn the intentions of the user by interpreting their actions as either *exploratory* or *expressive* [10]. Previous work has shown that coupling OLI with data visualization is helpful for data exploration [13].

A common method of presenting information to support OLI is with node-link diagrams. Using this technique, data

points are encoded in nodes, with the relationships between them encoded in the links. Using dimension reduction techniques such as Weighted Multidimensional Scaling [15] to compute link lengths based on the attributes of items, the distance between these nodes will reflect the relative similarity of each pair of data items. A group of nodes with many similar attributes will naturally begin to form implicit clusters in the data layout.

Indications and feedback from previous studies [1, 9] have shown that users favor clusters in the sensemaking process. These include systems with explicit clusters that are created to spatially organize information on the display (for example, iVisClustering [17]) and those that naturally develop due to the underlying layout updating through expressive interactions (for example, Andromeda [21]). It is therefore surprising that clusters are not explicitly defined in many OLI systems; rather, they implicitly exist in the layout structure. We are interested in exploring this new design space, to learn how to best integrate clustering algorithms into the active research area at the intersection of dimension reduction and OLI.

In this work, we propose, implement, and evaluate a similar model designed for quantitative data, which explicitly defines clusters within the layout of the data. In this model, the direct manipulation of nodes internal to these clusters represents the *exploratory* OLI interactions, while manipulations into and out of clusters represents *expressive* interactions. Through the expressive interactions, a feedback loop develops between the user and the layout algorithm. User manipulations with nodes and clusters suggest intentions regarding the importance of data attributes to the layout algorithm, which in turn updates the layout algorithm and hence the clusters with this information. Iterating through this sensemaking feedback loop provides insight to the user.

In addition, explicitly defining the clusters in this model allows us to easily solve the OLI “with respect to what” problem (discussed in [21]). In a visualization with dozens or hundreds of points, it is difficult to identify what each expressive data manipulation is moving with respect to: is the user moving the current node away from some set of points (and which points), or towards some set of points (and which points), or is the user merely organizing the node in an exploratory interaction? To solve this problem, we use clusters as the primary metaphor for interaction, limiting our expressive interactions to only include cluster membership changes: adding and removing nodes from clusters.

As OLI has evolved from its introduction [10, 18], the sensemaking pipeline has grown from a statistical model [8]

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

HILDA '17 Chicago, Illinois, USA

© 2017 ACM. ISBN 978-1-4503-5029-7/17/05...\$15.00

DOI: <http://dx.doi.org/10.1145/3077257.3077259>

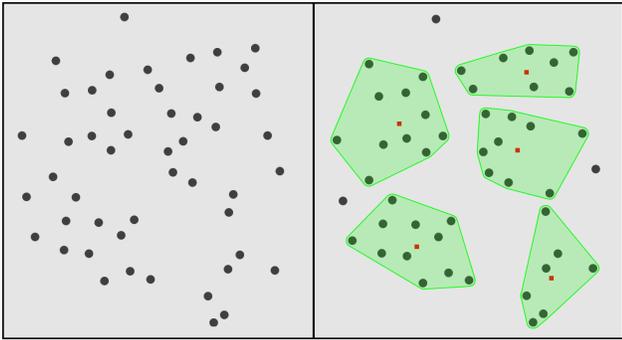


Figure 1: A comparison of the layout of the Case Study dataset both without (left) and with (right) clusters defined. Without clusters, the user can interact with the points to learn information about the properties of the animals (collie and chihuahua are close together, blue whale and giant panda are not), but are generally presented with a distribution of points without structure. With clusters, more information is readily provided to the user regarding the groups that nodes fall into.

to more complex multi-model pipelines [4] and systems combining OLI with parametric interaction [21]. Our contribution in this work is the proposal, implementation, and evaluation of this cluster-based OLI model. Though previous works include explicit data clusters and OLI interactions, this is the first system to explicitly define clusters and cluster interactions for OLI. Our intention here is to explore one solution in the clustering/dimension reduction/OLI design space as a stepping stone towards exploring the overall design space.

2. RELATED WORK

A common method of sensemaking with multidimensional data is to project the dataset into two dimensions. With this scatterplot or node-link diagram, users can draw conclusions from relative data positions. User manipulation of data items can be used to learn the distances between nodes, as shown by Dis-Function [5]. Data projection systems can be grouped into two collections: those which do and do not explicitly define clusters.

In the realm of explicit cluster definition, the iVisClustering system [17] includes a Cluster Relation View which encodes cluster membership using node color in a graph representation. TIARA [19] likewise clusters related text keywords into topical collections with importance changing across time, visually encoded in a streamgraph. Similarly, the Galaxy View in the IN-SPIRE system [23] projects the relationships between text documents into a two-dimensional space, with topic clusters defined. iCluster [20] organizes peers in peer-to-peer networks with common interests into clusters. Vizster [12] defines explicit clusters in the form of community structures in social networks. No OLI-style interactions are supported in these systems; the data is merely projected.

In contrast to explicit clustering systems, Andromeda [21] projects high-dimensional data into two dimensions using a method combining OLI with parametric interaction. Though similar data points implicitly cluster into groups, explicit

clusters are not defined. Similarly, StarSPIRE [4] enables OLI interactions on document collections that are projected into a two dimensional space, allowing similar documents to implicitly cluster together. In this system, the distance between nodes represents relatedness. iPCA [14] includes a projection view in which data items are projected onto two user-selected principle components. Apparent clusters can develop in the data in each of these systems, but are not explicitly defined.

Interactive clustering is also an active research area, with a wide variety of techniques to address user interaction with clusters. Chuang and Hsu provide a survey of human centered interactive clustering, listing a set of properties for effective interactive clustering algorithms [6]. Both Bilenko et al. [3] and Cohn and Caruana [7] explore the area of semi-supervised clustering, in which the user iteratively provides classification information to a machine learning algorithm that learns improved clusters with each iteration. A similar technique is used by Basu et al. in their interactive clustering system, allowing users to create explicit groups of items and receive suggestions for new items to add to each group [2]. Stahnke et al. introduce a set of “probing” interactions to explore projections of high-dimensional data in order to allow users to better interpret the meaning and quality of the visualization.

3. MODEL AND IMPLEMENTATION

Our model of this cluster-based OLI framework is shown in Figure 2. This bidirectional pipeline is divided into forward and backward directions, where the forward direction handles the data projection and the backward direction responds to user input. The following implementation makes use of the Euclidean distance function, a force-directed layout for dimension reduction, and k-means clustering; however, the model generalizes to any distance function, dimension reduction technique, and clustering algorithm.

3.1 Forward Direction

Distance Function The forward direction for data projection begins with computing a distance $\delta(n_i, n_j)$ between each pair of data items n_i and n_j . This distance, described in Equation 1, is the Euclidean or L_2 distance between the normalized attributes of n_i and n_j , including a weight w_a applied to each attribute a to denote the importance of that attribute to the current projection. At system initialization, each of the weights associated with the dimensions in the dataset are set to 1, indicating that each weight has no larger or smaller effect on the resting length of each link than any other weight. These weights are updated in response to user interaction in the backward pipeline direction to create new projections, detailed in the next subsection.

$$\delta(n_i, n_j) = \sqrt{\sum_{a \in attr} w_a * (n_{i,a} - n_{j,a})^2} \quad (1)$$

Force-Directed Layout Once a distance is computed for every data pair, we load the data into a force-directed node-link visualization such that every data item n_i is encoded as a node and every distance $d(n_i, n_j)$ between pairs of data items is encoded as a link with the distance value mapped to the resting length of the link. To address the nondeterministic layout challenge inherent to force-directed placement, node positions are initially set to the same loca-

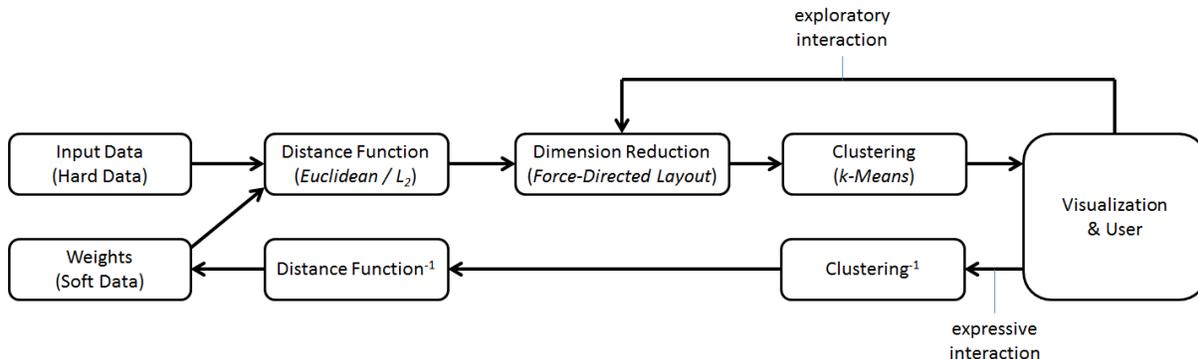


Figure 2: The framework and implementation of our cluster-based OLI model. Moving forward (right) along the pipeline, data is represented in a node-link diagram where link lengths encode similarity. A force-directed algorithm places the nodes to minimize force exerted by the links as they converge to their resting weights, while a modified k-means algorithm determines clusters in the data. Users then interact with the visualization, either through *exploratory* interactions which solely adjust the visualization (the force directed layout and forward) or through *expressive* interactions which follow the pipeline backward. In this case, the system determines the clusters affected by the interaction, computes similarity measures to learn which weights to update, and changes the weights associated with attributes in the layout, leading to the next forward flow along the pipeline. Though we use Euclidean distance, a force-directed layout, and k-means clustering, the model generalizes to any distance function, dimension reduction technique, and clustering algorithm.

tion every time the system is initialized. The nodes begin at locations uniformly and radially spaced about the center of the display. The force-directed layout algorithm is run repeatedly until it converges to a relatively stable layout, at which time we begin to visualize the layout.

Clustering As the force-directed layout converges to a solution, we begin to draw clusters surrounding groups of nodes. These clusters are computed using a modified k-means algorithm, which has been altered to include a maximum cluster radius that allows some nodes to exist external to all clusters. Similarly to the initial node placements, cluster centroid positions are initialized uniformly and radially spaced about the center of the display, and converge towards a final cluster layout as the force-directed algorithm continues to update the node positions. The k-means clusters are kept stable by seeding the new clusters with the previous centroid positions each time the display refreshes. We selected k-means as an efficient and simple clustering algorithm. The user is afforded control of both k and the size of each cluster in pixels from the centroid (the system defaults are $k = 5$ clusters and a 200 pixel radius for each cluster). Clusters are drawn using the Graham scan algorithm for convex hulls [11].

Visualization & User The user does not need to wait until a final layout of nodes and clusters is reached before beginning to interact with the system. They can begin performing interactions from the moment that the layout begins to render on the display, which triggers the backward direction of the pipeline as described in the next subsection.

3.2 Backward Direction

The user interacts with the nodes via direct manipulation, using click-and-drag actions to move nodes around the screen and mouse over interactions to see the details of a node. Keyboard inputs allow the user to change the number and size of clusters. As noted in the introduction, users can perform both *exploratory* and *expressive* interactions in

OLI systems. These interactions take two different backward paths through the pipeline.

Exploratory interactions are defined as interactions that alter the layout of the projection but do not affect the model. In contrast, expressive interactions will alter the model, in turn also affecting the layout of the projection. To address the “with respect to what” OLI challenge, we separate exploratory and expressive interactions based upon their effects on the clusters.

Exploratory interactions are interactions with nodes in which the user does not move a node into or out of a cluster; in other words, moving a node internal to a cluster to a new location in the same cluster, or moving a node external to all clusters to a new location external to all clusters. We note that it is possible that an exploratory interaction will cause nodes to shift membership of nodes between clusters in response to this interaction. However, these membership changes are due to the force-directed layout rather than any change in underlying model parameters. Additionally, moving nodes will pin their location on the display, which forces a further alteration to the force-directed layout so that links support the new position of the node. Again, this does not change the distance model parameters.

Clustering⁻¹ We define three expressive interactions that update weights in the graph: adding a node from the external region to a cluster, removing a node from a cluster to the external region, and transferring a node from one cluster into another. The third interaction, transferring a node between clusters, is interpreted as a sequence of the first two actions, removing the node from the first cluster and then adding it to the second cluster. Thus, we will discuss only the insertion and removal processes in detail. The goal of each of these interactions is to learn the projection and clustering parameters based on the intent of the cluster assignments that the user performs, thus creating a new projection and set of clusters that reflect the user action.

3.2.1 Adding Nodes to Clusters

Distance Function⁻¹ If a node is moved from the external region into a cluster, we aim to understand why the user decided that the node belongs to this cluster by analyzing what attributes in this node are similar to those attributes present in the cluster, and also what attributes in this node conflicts with in the cluster. To make this judgment, we use a heuristic approach to efficiently approximate an inversion of the distance function. We compare each attribute a of the cluster centroid c with the corresponding attribute of the newly added node n . This comparison is a calculation similar to that of our initial distance computation, normalizing the difference in value for each attribute between node and the cluster centroid, and is shown is Equation 2. Note that here we use L_1 or Manhattan distance rather than Euclidean distance, because we consider each attribute independently with the goal of sorting them rather than considering the attributes collectively to calculate an overall distance. The motivation behind this computation comes from the user interaction – the user has decided that node n belongs to cluster c , and so the attributes that are most similar between the node and the cluster are important to the user. Therefore, the model should reflect the importance of these attributes in the visualization. Similarly, the attributes that are most different between node and cluster are irrelevant to the user and hence less important to the model.

$$\forall a \in attr, \delta(c_a, n_a) = |c_a - n_a| \quad (2)$$

After this similarity analysis is complete for each attribute, we sort the attribute collection based on the strength of similarity score computed, with the sorted positions of attributes with tied similarity scores placed arbitrarily. A function is then applied to each of these sorted attributes to update the weights of each of the attributes, so that attributes that show the greatest similarity pull nodes closer together while attributes that do not push nodes further apart. The attributes that display strong similarities between the node and the cluster have their weights reduced, so that when the link lengths are recalculated, nodes that have similar values for this attribute are pulled closer together. Similarly, attributes displaying difference between the node and the cluster have their weights increased to expand link lengths after the recalculation phase. Attributes near the middle of the pack have little change, with weight updates close to 1.

Following the weight updates, the system proceeds through the forward direction of the pipeline again, recomputing distances and resting lengths and updating the layout and clusters accordingly. As the layout is stabilizing, new insights can be drawn about the properties and layout of the nodes, and the system is ready for new interactions.

We note that it is possible for a note already positioned inside of a cluster to be forced out of the cluster because of a subsequent interaction. This is both because the system learns gradually rather than immediately and because we do not enforce must-belong and must-not-belong cluster membership constraints unless the nodes are pinned in place. A number of user-driven cluster assignments are required for the system to converge to a user-intended ideal projection. We chose to allow the system to update after every user interaction (rather than to allow a number of interactions followed by an “update layout” trigger) to allow users to immediately begin to learn and draw insights from individual interactions.

3.2.2 Removing Nodes from Clusters

Distance Function⁻¹ If a node is dragged from a cluster into the external region between clusters, we aim to understand why the user decided that this node does not belong to its assigned cluster through a similar heuristic to the Add Node procedure. Again, we compare the attributes of the centroid of the cluster with the attributes of the removed node and sort the scaled outcomes.

Following this sorting, we again update the weights on the attribute collection based on the order of similarity scores. However, attributes that display strong similarities between the node and the cluster now have their weights increased to expand the length of links (interpreting formerly strong similarities as attributes that the user is not interested in clustering), while attributes that display differences between the node and the cluster have their weights reduced to shrink links (noting that this attribute is more important to the user than it was in the previous layout). Again, after all weights have been updated, new resting lengths are computed for each link, and the force-directed algorithm shuffles nodes into a new layout, with cluster membership updating as necessary.

4. CASE STUDY

Here, we show one example of exploring a dataset via cluster-based OLI. The dataset used in this case study is a collection of 49 animals, each defined by 85 numeric attributes that describe the animal’s color, physical characteristics, habitat, and behavior [16]. The data is loaded into the system using the default settings, creating five clusters with cluster membership size set at 200 pixels from the centroid. These clusters, along with the user actions described below, are shown in Figure 3.

Before any user interactions take place, the similarity relationships already in the data have clustered 46 of the animals into groups with similar characteristics. Clockwise from the top, these groups can be summarized as predators (including animals such as the lion and tiger), pets (chihuahua and Persian cat), aquatic animals (dolphin and seal), large grazers (buffalo and sheep), and small foragers (skunk and rabbit). Three animals (bat, polar bear, zebra) were not immediately assigned to any cluster. There are also several animals that are misclassified according to this interpretation of the clusters, including the grizzly bear in the pets cluster and the giant panda in the small foragers cluster. For the purposes of clarity and mental map stability in this case study, we perform the exploratory action of fixing one node from each cluster in place, an action that has no affect on the underlying weights.

In exploring the data, one might begin by resolving the pet grizzly bear issue by removing the grizzly bear node from the pet cluster and placing it in the predator cluster. This combination of two cluster transfers (out of pet, into predator) is an expressive action that results in a weight update. Following this action, the “fierce” weight has dropped from 1.000 to 0.437, indicating importance to that attribute in the current exploration and pulling nodes closer together through its impact on the resting length of all edges. In contrast, the “orange” color weight has increased from 1.000 to 1.651, indicating a lack of importance to that attribute and increasing the contribution of that attribute to the rest lengths of all links.

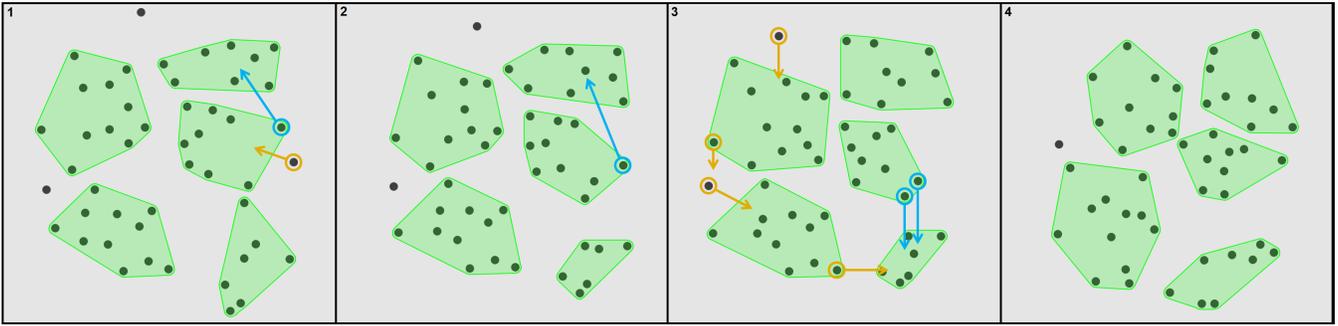


Figure 3: Four steps through the interaction discussed in the Case Study section. In Frame 1, the system is initialized with 5 clusters (green convex hulls). The user drags the grizzly bear node into the predator cluster (blue arrow), which updates the weights and pulls the polar bear node into the pet cluster (orange arrow). In Frame 2, the user corrects this by putting the polar bear into the predator cluster (blue arrow). In Frame 3, the user drags the otter and beaver nodes into the aquatic cluster (blue arrows), which cause the bat to join the small foragers, the zebra to join the large grazers, the giant panda to escape the small foragers and travel towards but not yet into the large grazers, and the walrus to transfer from the large grazers to the aquatic cluster (orange arrows). Frame 4 shows the final alignment.

This action has absorbed the polar bear node into the pet class incorrectly, so the user can also move the polar bear node into the predator cluster, an expressive action that further reduces the “fierce” weight to 0.224. Because of these actions, the otter has now moved into the pet cluster and the beaver was already there, so the user may perform expressive actions to drag those into the aquatic animals cluster. After those interactions have completed, the “water” weight has decreased to 0.263, and the “fierce” weight has decreased further to 0.116. Other attributes that the model found important include 0.103 for meat-based diets and 0.531 for having hooves, while attributes such as nocturnal (2.259) and hairless (2.327) were not judged as important to the current exploration of the user.

Most importantly, without user interaction on these nodes, these actions had the effects of putting the zebra into the large grazers cluster, removing the giant panda from of the small forager cluster and moving towards large grazers, added the bat into the small grazers, and moved the walrus from large grazers to aquatic animals. The final state is shown in the right-most panel of Figure 3.

5. DISCUSSION

A number of the weight updates and animal shifts through the interactions in the case study are worth noting. For example, the zebra transitioned into the large grazers cluster and the giant panda left the small foragers cluster despite no user interaction with these nodes. The weight that the system judged to be most important was neither “water” nor “fierce” but “domestic,” with a 0.050 weight. Still, similar weights to “fierce” such as “hunter” were also reduced to small values (0.184). The weights judged least important to the user interactions are “slow” (2.435), “fast” (2.318), and “active” (2.326), demonstrating that the system learned that the user was interested in the diet and habitat of these animals, not their speed.

It is also interesting to think about the relationship between clustering algorithms and dimension reduction algorithms. In a way, making cluster assignments is equivalent to 1D dimension reduction, noting that the cluster assign-

ment is the primary dimension of organization. Our system goes a bit beyond that, still respecting the 2D position of the force-directed layout while also encoding an extra dimension of cluster membership.

5.1 Future Work

We have several targets for future work in OLI clustering. At the moment, our clusters are computed using k-means based on the Euclidean distance between nodes in the two-dimensional projection of the data laid out by the force-directed algorithm. Although visual clutter is more likely with the possibility of clusters overlapping, we would like to explore the idea of creating the clusters in high-dimensional space and then projecting them into 2D, to understand if different insights are drawn by switching the order of these operations.

We would also like to include interactions on the clusters themselves, supporting cluster mergers and cluster splits without needing to individually drag each node to its target. Collapsing clusters to a single glyph will help lead towards OLI on bigger datasets, as more data can be included in the available screen space. Adding additional visual cues to the clusters and nodes themselves would make it system actions such as a node exiting or entering a cluster more evident to users. For example, briefly making the node color flash when the giant panda node left the small foragers cluster. Finally, we would like to update the back-end model of the system from the heuristic weight-based method to a true layout model inversion.

5.2 Limitations

One limitation to the implementation of this framework is the solution we implemented to address the nondeterministic layout challenge: we always initialize nodes and centroids in the same locations. Because of this, the same clusters will develop every time, potentially biasing the user towards these clusters and hindering new directions of exploration. Altering this initial placement to be more random is a trivial solution to this source of bias.

Additionally, we currently leave both the choice of the k parameter and the radius threshold of each cluster as user-

defined parameters. The system could certainly make use of automated algorithms such as the elbow method [22] to determine the optimal number and size of clusters. We left control of these parameters with the user in order to afford as much freedom in the data exploration as possible.

6. CONCLUSION

In this work, we have proposed a cluster-based framework for OLI interactions. Our implementation of this framework includes a feedback loop between the user and the back-end model, in which the layout of the nodes informs the clusters, while the user interactions between the nodes and clusters update the weights that compute the layout. Though our specific implementation makes use of the Euclidean distance function, a force-directed layout, and k-means clustering, the overall model generalizes to any distance function, dimension reduction technique, and clustering algorithm. We show in a case study an example of sensemaking using a dataset of animal characteristics, demonstrating that the model can learn from user interactions and affect the layout of nodes and clusters that are not interacted with directly by the user.

7. ACKNOWLEDGMENTS

This research was supported by NSF Grant IIS-1447416 and by a grant from General Dynamics Mission Systems.

8. REFERENCES

- [1] C. Andrews, A. Endert, and C. North. Space to think: large high-resolution displays for sensemaking. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 55–64. ACM, 2010.
- [2] S. Basu, D. Fisher, S. M. Drucker, and H. Lu. Assisting users with clustering tasks by combining metric learning and classification. In *AAAI*, 2010.
- [3] M. Bilenko, S. Basu, and R. J. Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *Proceedings of the twenty-first international conference on Machine learning*, page 11. ACM, 2004.
- [4] L. Bradel, C. North, L. House, and S. Leman. Multi-model semantic interaction for text analytics. In *2014 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 163–172, Oct 2014.
- [5] E. T. Brown, J. Liu, C. E. Brodley, and R. Chang. Dis-function: Learning distance functions interactively. In *2012 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 83–92, Oct 2012.
- [6] J. Chuang and D. J. Hsu. Human-centered interactive clustering for data analysis. *Conference on Neural Information Processing Systems (NIPS). Workshop on Human-Propelled Machine Learning*, 2014.
- [7] D. Cohn and R. Caruana. Semi-supervised clustering: Incorporating user feedback to improve cluster utility. In *Proceedings of the Conf. on Artificial Intelligence*, 2000.
- [8] A. Endert, P. Fiaux, and C. North. Semantic interactions for visual text analytics. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '12*, pages 473–482, New York, NY, USA, 2012. ACM.
- [9] A. Endert, S. Fox, D. Maiti, and C. North. The semantics of clustering: analysis of user-generated spatializations of text documents. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, pages 555–562. ACM, 2012.
- [10] A. Endert, C. Han, D. Maiti, L. House, S. Leman, and C. North. Observation-level interaction with statistical models for visual analytics. In *2011 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 121–130, Oct 2011.
- [11] R. L. Graham and F. F. Yao. Finding the convex hull of a simple polygon. *Journal of Algorithms*, 4(4):324–331, 1983.
- [12] J. Heer and D. Boyd. Vizster: visualizing online social networks. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.*, pages 32–39, Oct 2005.
- [13] L. House and C. Han. Bayesian visual analytics: Bava. *Statistical Analysis and Data Mining*, 8(1):1–13, 2015.
- [14] D. H. Jeong, C. Ziemkiewicz, B. Fisher, W. Ribarsky, and R. Chang. ipca: An interactive system for pca-based visual analytics. In *Computer Graphics Forum*, volume 28, pages 767–774. Wiley Online Library, 2009.
- [15] J. B. Kruskal and M. Wish. *Multidimensional scaling*, volume 11. Sage, 1978.
- [16] C. H. Lampert, H. Nickisch, S. Harmeling, and J. Weidmann. Animals with attributes: A dataset for attribute based classification, 2009.
- [17] H. Lee, J. Kihm, J. Choo, J. Stasko, and H. Park. ivisclustering: An interactive visual document clustering via topic modeling. In *Computer Graphics Forum*, volume 31, pages 1155–1164. Wiley Online Library, 2012.
- [18] S. C. Leman, L. House, D. Maiti, A. Endert, and C. North. Visual to parametric interaction (V2PI). *PloS one*, 8(3):e50474, 2013.
- [19] S. Liu, M. X. Zhou, S. Pan, W. Qian, W. Cai, and X. Lian. Interactive, topic-based visual text summarization and analysis. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 543–552. ACM, 2009.
- [20] P. Raftopoulou and E. G. Petrakis. icluster: a self-organizing overlay network for p2p information retrieval. In *European Conference on Information Retrieval*, pages 65–76. Springer, 2008.
- [21] J. Z. Self, R. K. Vinayagam, J. T. Fry, and C. North. Bridging the gap between user intention and model parameters for human-in-the-loop data analytics. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics, HILDA '16*, pages 3:1–3:6, New York, NY, USA, 2016. ACM.
- [22] R. L. Thorndike. Who belongs in the family? *Psychometrika*, 18(4):267–276, 1953.
- [23] P. C. Wong, E. G. Hetzler, C. Posse, M. A. Whiting, S. Havre, N. Cramer, A. R. Shah, M. Singhal, A. Turner, and J. Thomas. In-spire infovis 2004 contest entry. In *INFOVIS*, volume 4, pages 51–52, 2004.