Pollux: Interactive Cluster-First Projections of High-Dimensional Data

John Wenskovitch* and Chris North†
Virginia Tech Discovery Analytics Center

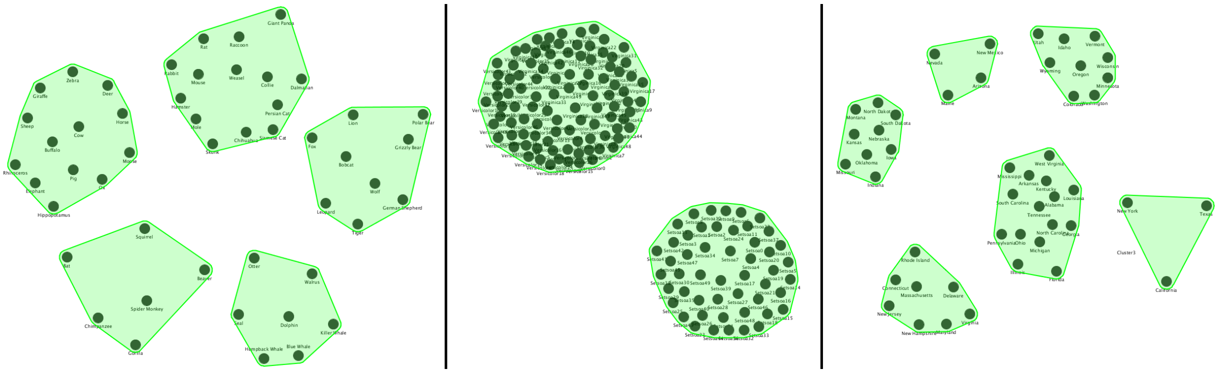

Figure 1: Clustered projections of three datasets generated by Pollux. From left to right, an Animals dataset [37], the Fisher Iris dataset [18], and a U.S. Census States dataset [53].

Abstract

Semantic interaction is a technique relying upon the interactive semantic exploration of data. When an analyst manipulates data items within a visualization, an underlying model learns from the intent underlying these interactions, updating the parameters of the model controlling the visualization. In this work, we propose, implement, and evaluate a model which defines clusters within this data projection, then projects these clusters into a two-dimensional space using a "proximity≈similarity" metaphor. These clusters act as targets against which data values can be manipulated, providing explicit user-driven cluster membership assignments to train the underlying models. Using this cluster-first approach can improve the speed and efficiency of laying out a projection of high-dimensional data, with the tradeoff of distorting the global projection space.

Keywords: Dimension reduction, clustering, semantic interaction, exploratory data analysis.

Index Terms: Human-centered computing—Visualization—Visual analytics

1 Introduction

In recent years, analysts have worked to explore and draw conclusions from increasingly larger datasets. As a result, visual analytics tools continue to grow more complex, with computational pipelines that transform data into interactive visualizations often consisting of multiple analytical models. These multi-model systems are becoming prevalent, and include but are not limited to combinations such as relevance and similarity [7], sampling and projection [41], and most relevant to this work, dimension reduction and clustering [56].

Though dimension reduction and clustering algorithms serve different cognitive purposes in visualizations (spatializing and grouping, respectively), they can naturally coexist within a projection of data. Many dimension reduction algorithms make use of a "proximity ≈ similarity" metaphor; in other words, the similarity of observations in the high-dimensional space is mapped to the proximity of the corresponding nodes in the low-dimensional projection [38]. As a result, groups of similar items form *implicit* clusters in the projection. If these clusters are defined *explicitly* by a clustering algorithm, additional structural information about the data can be communicated to the analyst. That said, clusters are inherently subjective structures, a fact that makes their identification a challenging process for both the analyst and the machine [27].

In our previous work, we explored the combination of dimension reduction and clustering algorithms within visual analytics systems [55], identifying the various ways of combining these two algorithm families into the same system. However, few implementations make use of semantic interaction and similar learning routines to infer the intent of an analyst and train a model, instead relying on direct parametric feedback via user interactions. In other previous work, we implemented Castor [56], a tool that first reduces the dimensionality of the data from the high-dimensional input into a two-dimensional projection, and then runs the clustering algorithm on that reduced data. As an analyst updates the cluster memberships of individual observations, the system gradually learns which dimensions are of most interest to the analyst's current exploration, and Castor updates the projection to better reflect that interest.

In this work, we introduce Pollux. Pollux is similar to Castor, but the algorithm order is reversed: the data is first clustered in the high-dimensional space, and is then projected into a two-dimensional visualization. Such a process has previously been included in analytical tools [14], but these projects did not include semantic interaction learning as seen in Castor. Introducing online learning into Pollux permits analysts to maintain a data exploration focus, with no need for mental context switching to ponder model parameters. Determining how to present the outcome of this computational flow leads to a number of design options that can be considered, reflecting a balance between an accurate projection of the data and faster rendering.

In particular, we note the following contributions:

1. The design and implementation of Pollux, an interactive cluster-first system that learns observation classifications via analyst feedback and displays using a unified layout model based on edge classes.
2. A discussion of the benefits of the cluster-first model, as well as of methods that can extend the cluster-first design space beyond that which has been implemented in this work.

*jw87@cs.vt.edu
†north@cs.vt.edu

2 Related Work

2.1 Interactive Dimension Reduction

The semantic interaction work initiated by Endert et al. [22–26] has led to much of the current research into interactive dimension reduction tools. Under this semantic interaction paradigm, incremental feedback is delivered to the system by an analyst [48]. The system then uses these interactions to infer the intent of the analyst and to update the projection accordingly, often via adjustments to a vector of weights applied to the dimensions of the dataset.

These incremental dimension reduction tools can be divided into classes that support quantitative and text data. Quantitative data is straightforward, as dimension reduction algorithms process numerical data and distances by default. Tools such as Andromeda [46,47] are built on the V2PI framework [39] for parameteritizing and learning from user interactions, and can be supplemented with supporting views, as seen in Dis-Function [8]. Alternative frameworks and learning approaches are seen in the LAMP framework from Joia et al. [34] and the iLAMP extension [16], the Piecewise Laplacian technique described by Paulovich et al. [44], and the tools developed by Mamani et al. [41] and Molchanov et al [43]. ModelSpace [9] visualizes interaction provenance trails, projecting high-dimensional vectors to show the exploration strategies of Dis-Function and Doc-Function users.

The text data case is a special variant of the quantitative case, as text must be processed into numerical data before running the dimension reduction algorithms. These numerical values often take the form of term frequencies, or term frequencies scaled by the frequency of that term appearing in the overall corpus (TF-IDF). However, the sparcity of the resulting data requires different processing techniques, such as replacing the common Euclidean or Manhattan methods for measuring the distance between observations with either Cosine Distance to handle the sparcity [49] or with Gower distance to handle missing attributes [28]. Tools such as StarSPIRE [7,54] and Cosmos [17] allow for the interactive exploration of document collections, with StarSPIRE opting for a force-directed layout and Cosmos using WMDS.

2.2 Interactive Clustering

Interactive clustering serves several purposes for the analysis of data, depending on the goals of the analyst. Typically, the analyst wishes to find the clustering assignment that best suits their current search strategy or supports their targeted conclusions [3,40]. Systems such as SOPHIA provide analysts with support for exploratory search and retrieval of documents, in this case for medical documents [15]. The goal of an ideal interactive clustering system is to understand these analysts and adapt the clustering to suit their intent [11,30,50]. Some systems provide analysts with options, displaying multiple clustering results and allowing the analyst to choose the best solution [21]. Still others aim to highlight interesting data automatically for the analyst, guiding their exploration to regions of the data [2,6]. Interactive topic modeling allows analysts to see groups of documents based on common topics of interest [21,32,33,35].

Interactive clustering systems require a set of interactions to receive feedback from analysts. At a basic level, an analyst can be given the ability to directly adjust the number of clusters created, or to directly modify parameters that control those clusters such as a distance threshold [6, 15, 30, 40, 50]. Systems can also support direct interactions with the clusters, such as merging and splitting clusters [6,10,32], removing clusters [5,15,30,40], and hiding and expanding clusters [3,5,6,15]. When looking at individual observations within the clusters, analysts are often afforded the ability to move nodes from one cluster to another [5,12], referred to by Dubey et al. as "Assignment Feedback" [19]. Through such interactions, analysts can supply must-link and cannot-link constraints to clustering solutions [8,33]. A unique feature of Pollux is the projection of relationships within and between clusters using a spatialization based on the clustering operations and assignments.

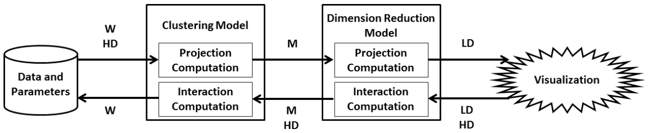

Figure 2: The computational pipeline for Pollux. The projection computations convert data into a visualization, while the interaction computations interpret and respond to analyst interactions.

3 Pollux

The goal with the Pollux system was to continue to explore the interaction space between dimension reduction and clustering algorithms [55] by introducing a cluster-first system. Pollux differs from many other interactive clustering systems because of the inclusion of projections via dimension reduction, displaying learned similarities at both the cluster and observation level through user-driven reclassification. An analyst using Pollux should be afforded the ability to update the system-learned categorization of observation, training the underlying clustering and dimension reduction models to better express their current exploration interests. Further, an analyst should be able to receive feedback from these algorithms concurrent with the incremental learning process, permitting the analyst to update or alter their exploration based on the most current results displayed.

Our model of this cluster-first framework is shown in Fig. 2. This bidirectional pipeline is divided into projection and interaction directions, where the projection direction converts input data into an interactive visualization, and the interaction direction responds to analyst input. These projections and interactions are supported by Dimension Reduction and Clustering Models, which work cooperatively to generate an interactive visualization from the provided high-dimensional dataset and a learned weight vector. The implementation described in this section 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 Projection Direction

At a high level, the projection direction computes clustering assignments for the high-dimensional observations, and then structures a visual representation of those clusters into a two-dimensional space. There are many methods to visually convey cluster membership, and again, we elected to follow the visual style of Castor for this application. A broader discussion of visualization structures and alternatives follows in Sect. 4.

Weighted Cluster Assignments In the projection direction of the Pollux pipeline, the Clustering Model is the first to execute. This cluster model has two primary goals: to determine a quality clustering assignment for the observations given the data at hand, and to communicate those membership assignments to the Dimension Reduction Model for layout.

To accomplish the first goal, a weighted k-means algorithm is executed on the dataset. In the default implementation of the system, we execute 500 versions of k-means for each value of k ranging from 2 to 15. Each of the best k-clusterings (as determined by summed intra-cluster distance) is stored, and an optimal k value is determined from these best clusterings using the elbow method [51]. The analyst is afforded control of k, so that they can refine the number of clusters generated by the system if the initially-selected version does not suit their goals. After each of the clusters has been determined, an additional node is created to specifically represent the centroid of the cluster.

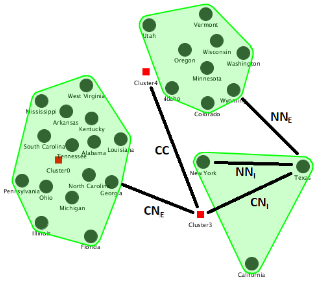

Figure 3: Five different classes of edges that could be included in the layout: Centroid-Centroid Edges (CC), Centroid-Node Internal Edges ($\mathrm{CN_I}$), Node-Node Internal Edges ($\mathrm{NN_I}$), Centroid-Node External Edges ($\mathrm{CN_E}$), and Node-Node External Edges ($\mathrm{NN_E}$).

Projecting Clusters After cluster memberships have been determined, the Dimension Reduction Model is tasked with projecting these clusters into the visualization. The precise layout of this visualization is dependent upon the importance of each class of edges that is included in the layout. There are five of these classes, shown in Fig. 3:

- *Centroid-Centroid Edges* (CC): Distances between the clusters themselves, displaying the similarity between pairs of clusters.
- *Centroid-Node Internal Edges* ($\mathrm{CN_I}$): Distances between each cluster member and its centroid, demonstrating the centrality of a node in the cluster. These edges act to pull associated nodes towards their cluster centroid.
- *Node-Node Internal Edges* ($\mathrm{NN_I}$): Distances internally between cluster members. These edges display the similarity of nodes within a single cluster, providing an overall organizational structure to the members of a cluster.
- *Centroid-Node External Edges* ($\mathrm{CN_E}$): Distances between nodes and the centroids of other clusters. These edges pull nodes within a cluster towards the direction of alternative cluster memberships.
- *Node-Node External Edges* ($\mathrm{NN_E}$): Distances globally between observations, with no regard for cluster boundaries (but not including edges internal to a cluster). These edges pull nodes directly towards similar observations in other clusters, showing pairwise relationships between observations.

The classes of edges that are included in the visualization impact both the accuracy and rendering speed of the visualization. A longer discussion of this tradeoff is included in Sec. 4.

After edges are constructed in the graph, a distance $\delta(n_i, n_j)$ is computed for every pair of nodes and centroids with a connecting edge. This distance, described in Equation 1, is the L_2 or Euclidean distance between the normalized attributes of endpoints n_i and n_j, including an attribute weight w_a applied to each attribute a to denote the importance of the associated dimension to the current projection. At system initialization, each of these weights 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 attribute weights are updated in response to analyst interactions in the interaction direction, detailed in the next subsection. A further edge class weight, w_e, is applied to each of the edges. This edge class weight allows for different styles of visualization to be created (e.g., compact clusters, tightly grouped clusters, broad clusters). Tradeoffs in this design space are also discussed in more detail in Sec. 4.3. These computed edge lengths are then treated as the optimal resting lengths within a force-directed simulation, with nodes beginning at locations uniformly and radially spaced about the center of the display and updating their positions until the layout converges to a relatively stable layout. Clusters are drawn using the Graham scan algorithm for convex hulls [29].

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

3.2 Interaction Direction

The goal of the interaction direction is to respond to analyst interactions, incrementally training the underlying models and learning the intent of the analyst when they perform reclassification interactions. The analyst interacts with the nodes via direct manipulation, using click-and-drag actions to move nodes between clusters. Mouseover interactions afford a details-on-demand view of the raw data for each node. Analysts have the ability to perform two types of interactions, each of which are addressed by a different model in Pollux.

Layout Interactions These interactions are addressed by the Dimension Reduction Model, as no clustering updates need to be performed. Such interactions can be used to probe relationships internal to a cluster, perhaps dragging a node from one side of the cluster to the other and watching the updates to the rest of the layout. These interactions can also assist the force-directed optimization in Pollux to shift between various local minima in the layout of observations, and could be extended to navigating the rotation and scale invariance properties of other dimension reduction algorithms such as MDS. Performing such interactions will not trigger the learning of new attribute weights; these are only learned via expressive interactions.

Cluster Interactions These interactions are performed when an analyst reclassifies an observation, dragging it from one cluster into another. As this interaction appears to demonstrate an analyst's dissatisfaction with the automated membership assignment, the system begins to learn a distance metric that matches the current exploration interests of the analyst, inferring the semantic reasoning behind this reclassification by examining the attributes of the dragged node, the source cluster, and the destination cluster. These interactions train the Clustering Model via incremental feedback.

To make this judgment, we use an incremental metric learning approach to efficiently compute an updated distance function. We compare each attribute a of the source cluster centroid cs and the destination cluster centroid cd with the corresponding attribute of the dragged node n. As shown in Equations 2 and 3, this comparison is a calculation similar to that of our initial distance computation, normalizing the difference in value for each attribute between the node and cluster centroids. One important difference is that here we use L_1 or Manhattan distances 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.

$$\forall a \in attr, \delta(cs_a, n_a) = |cs_a - n_a| \qquad (2)$$

$$\forall a \in attr, \delta(cd_a, n_a) = |cd_a - n_a| \qquad (3)$$

After computing this similarity distance for each attribute, we sort the attribute collections based on the strength of similarity score computed, with the sorted positions of attributes with tied similarity scores placed arbitrarily. A linear function is then applied to each of these sorted attributes to update the weight of each attribute. Attributes that show the greatest similarity between cluster and node should pull the pair closer together and so the weight is reduced, while attributes that show the least similarity should push the pair apart and so the weight is increased. Attributes near the middle of the list have little weight, with weight updates only a fraction from 1. With these weight scaling factors set, we first apply the factors between source cluster and node, followed by those of the destination cluster and node.

After the attribute weights have been updated, the system must update the visualization through the projection direction of the pipeline again. First, cluster assignments for each node are recomputed with

Table 1: A summary of the three datasets visualized with Pollux, enumerating each edge type.

Dataset	**Animals** [37]	**Fisher's Iris** [53]	**Census** [18]
Nodes	49	48	150
Dimensions	85	35	4
Clusters	5	6	2
Fully-Connected Node Graph	1176	1128	11175
Centroid-Centroid Edges (CC)	10	15	1
Centroid-Node Internal Edges ($\mathbf{CN_I}$)	49	48	150
Node-Node Internal Edges ($\mathbf{NN_I}$)	241	214	6175
Centroid-Node External Edges ($\mathbf{CN_E}$)	196	240	150
Node-Node External Edges ($\mathbf{NN_E}$)	935	914	5000

the new weight information. Any node that receives a new cluster assignment will have its adjacent edges updated as needed, which could include the removal of unneeded edges, the introduction of new edges, or the weighting of an edge that has transitioned from internal to external or vice versa. The force-directed layout then executes, with nodes that switch clusters smoothly animating from source to destination cluster.

As the system is currently designed with a small set of supported interactions, the process of displaying a clustering projection and learning an updated distance function is fairly straightforward. However, future versions of the system can also make use of the various edge types in determining the distance function updates. For example, a Layout Interaction might update some properties of only the CN_I and NN_I edges. Other interactions such as dragging two entire clusters closer together may refine the impact of the CC edges.

4 Extended Design Space

As noted at the beginning of Sec. 3, the Pollux model can be generalized to any distance function, dimension reduction technique, and clustering algorithm. Castor [56] held the same property. However, there are some additional properties of the Pollux technique that enable additional variants to be created from this cluster-first approach. In particular, we discuss in this section the roles of edge type selection and edge class weights on the visualization that is created.

4.1 Edge Class Selection

As noted in Sec. 3.1, there are five different classes of edges that exist within a Pollux projection. The role of these edges in the force-directed layout is the same as the role of distances in many dimension-reduction projections: to communicate a measure of similarity between two objects in the visualization. With the added introduction of clustering into Pollux projections, layout time can be improved and visualizations can therefore be generated more rapidly than in pure projection applications.

Table 1 provides a summary of nodes, dimensions, clusters (as learned by Pollux), and edge counts for three different datasets: a dataset of animals and a collection of appearance, habitat, diet, and behavioral attributes [37]; the traditional Fisher's Iris dataset [18]; and a dataset of demographic, employment, and housing data for the 48 continental U.S. States [53]. In this table, the "Fully-Connected Node Graph" row provides the number of edges or distances required to lay out a fully-connected graph of only observations (this is merely the sum of the NN_I and NN_E categories). Each of these classes of edges communicate additional information to the analyst, as listed in Sec. 3.1.

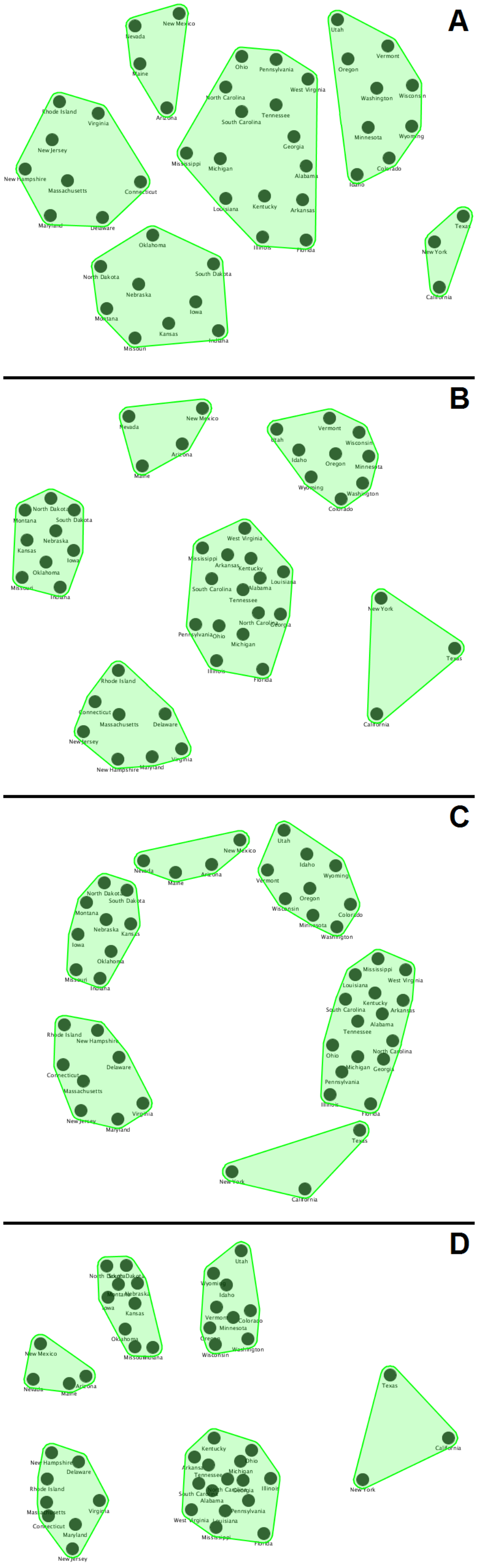

Figure 4: Four views of the Census dataset with a variety of edge class selections. From top to bottom, **(A)** only CC and CN_I edges, **(B)** same as above, plus NN_I, **(C)** same as above, plus CN_E, **(D)** all edge types.

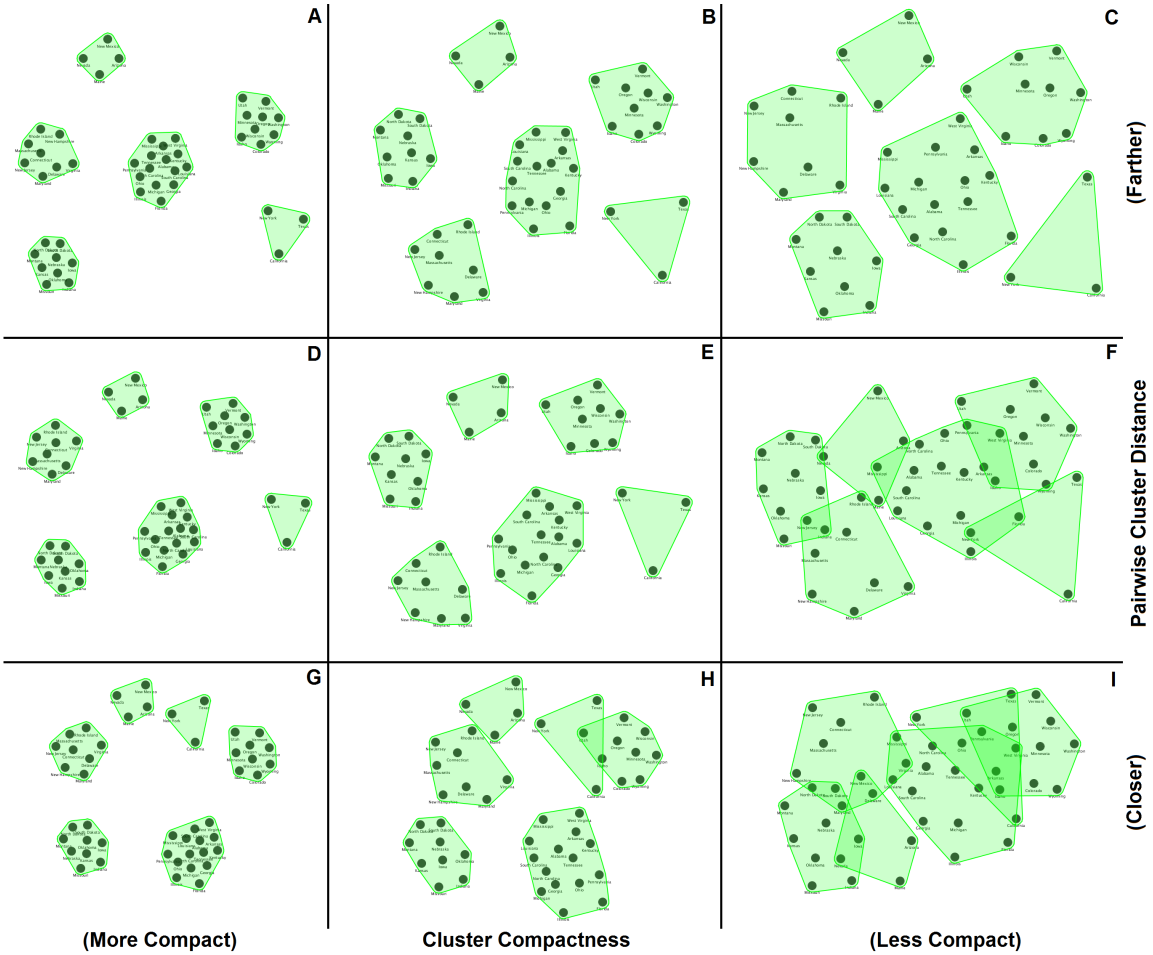

Figure 5: Nine views of the Census dataset with various CC, CN_I, and NN_I weights. Cluster compactness varies across the x-axis via manipulation of the CN_I and NN_I edge class weights, while pairwise cluster distance varies in the y-axis via manipulation of the CC edge class weight.

4.2 The Effect of Edge Class Selection on Performance

By selecting only the CC, CN_I, and NN_I edge classes, the number of edges that need to be computed for a projection is reduced by approximately 25-50% for these datasets, while still displaying the relative similarity of both clusters and observations internal to those clusters. Though force-directed simulations can run as quickly as the $O(n \log n)$ of the Barnes-Hut simulation approximation [4], many force-directed simulation implementations are $O(n^3)$, yielding a significant performance boost with a reduction to half of the original number of edges.

Figure 4 shows Pollux layouts with default edge class weights for the Census dataset, displaying the visualizations generated from four different edge class selections. In Fig. 4A, only the CC and CN_I edges have been selected. As a result, both pairwise cluster similarities (CC) and cluster memberships (CN_I) are displayed. In Fig. 4B, the NN_I have been added. The added attractive forces between nodes internal to each cluster act to compact the clusters in most cases while also incorporating pairwise node similarities internal to each cluster. Fig. 4C adds CN_E edges, which act to pull nodes within clusters towards the centroid of other clusters, thereby causing a more radial layout with the nodes on the inner ring boundary most attracted to other clusters and those on the outer ring boundary least attracted. Finally, Fig. 4D adds NN_E edges, which again cause the clusters to compact as many additional edges are added to the graph.

4.3 The Role of Edge Class Weights

In addition to selecting edge classes, each class is paired with an associated edge class weight w_e. The purpose of this weight is to influence the layout of the projection, allowing for Pollux to generate a variety of visual representations. Determining how to best lay out observations after performing cluster assignments is a dataset- and user-driven process, with the ideal layout of the data being influenced by the insight that the analyst wishes to communicate with the visualization. There exists a natural tradeoff that is implied through manipulating the layout between the distortion of the space and the best representation of these insights.

Fig. 5 shows several Pollux layouts for the Census dataset with varied of CC, CN_I, and NN_I edge class weight values. From left to right, the CN_I and NN_I weights are altered to change the compactness of the clusters. From top to bottom, the CC weight is altered to change the pairwise distances between the clusters. As clusters become less compact, they begin to overlap, causing some ambiguity in the cluster membership assignment of some nodes (e.g., Arkansas and Ohio in Fig. 5I). This effect is magnified as the relative pairwise distance between clusters is reduced.

4.4 Alternate Visual Representations

A limitation to the Pollux technique is the inherent space distortion required by the cluster-first projections. In other words, the Dimension Reduction Model in the Pollux pipeline assumes that cluster

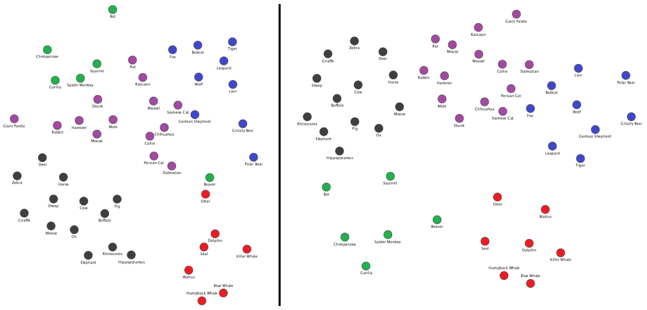

Figure 6: **(left)** A direct two-dimensional projection of the high-dimensional Animals data with cluster information encoded by color. **(right)** The same data in Pollux, using color encoding for clusters rather than convex hulls.

membership information from the Clustering Model is the primary layout factor, with weights a secondary factor and high-dimensional distances a tertiary factor. As a result, there is no way to avoid such spatial distortions without transforming the pipeline.

For example, the left panel of Fig. 6 depicts an alternative layout of the Animals dataset. In this layout, both the projection and the layout are computed from the high-dimensional data separately. The projection then accurately reflects all pairwise distances between observations, and the cluster membership is encoded by color rather than in convex hulls. However, this is not an instance of the Pollux pipeline; rather, it matches the "Independent Algorithms" pipeline identified in our taxonomy in previous work [55]. Additionally, the clusters are not as compact and easily identifiable in this view.

A Pollux representation of this Animals dataset using the same color mapping is provided in the right panel of Fig. 6. In this view, the clusters are more compact and uniformly-shaped, though the view without convex hulls may not clearly imply that there is no inter-cluster similarity at the node level in this view, as the CN_E and NN_E edges were not included when generating this projection.

4.5 Analyst Control of *k*

The Pollux examples provided thus far use the system-determined value of *k* to categorize and lay out the observations. However, the analyst is afforded with the ability to manipulate the value of *k* to update the cluster membership assignments. For example, the Fisher's Iris dataset consists of three different species of iris: Setosa, Virginica, and Versicolor; however, Pollux only determines that two clusters exist in the dataset, and does not differentiate between the Virginica and Versicolor species (Fig. 7 left). When the analyst updates the system to force three clusters, the Virginica and Versicolor species are separated, albeit imperfectly. The analyst can then begin to perform reclassification interactions to train the system to distinguish between Virginica and Versicolor.

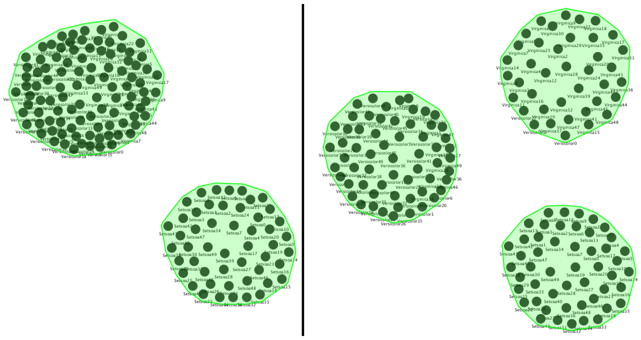

Figure 7: **(left)** The Fisher's Iris dataset with the system-determined two clusters. **(right)** The analyst updates the view to incorporate three clusters.

4.6 Extending the Hierarchy

Pollux as described thus far only consists of a single set of clusters containing nodes; however, the technique can be extended to include cluster hierarchies. Additional research is necessary to design interaction techniques for disambiguating between cluster reassignments in such a hierarchy. However, the benefit is the ability to visualize much larger datasets by also interactively expanding and contracting clusters. A similar technique was used by ASK-GraphView [1] to visualize hierarchically-clustered datasets several orders of magnitude larger than those demonstrated with Pollux in this work.

4.7 When to Learn?

As described in Sec. 3.2, Pollux contains two learning phases, identifying the reasoning behind the action of an analyst both removing an observation from a cluster as well as inserting the observation into a different cluster. However, there may be cases when the source cluster is not meaningful (an associated analyst intent might be "The Fox should be in the Predators cluster") and cases when the target cluster is not meaningful ("Alaska does not belong in the cluster of high-population states"). In these cases, only one learning phases is relevant to the interaction, and thus the second learning phase captures a portion of the interaction that has no associated analyst intent. The attribute weights are therefore updated needlessly. A second component to the interaction could help to determine which portions of the interactions have meaning. For example, a click-and-hold interaction before dragging could indicate that the removal from the source cluster is meaningful, while holding the mouse button down for a short period before releasing could indicate that insertion into the target cluster in meaningful. Again, additional research is necessary to design the best interaction technique.

Disambiguating cases where only the source cluster is important, only the target cluster is important, and cases where both are important is related to the "With Respect to What" problem detailed by Self et al [47]. The original definition of this problem was focused on disambiguation of intent between interactions relationships, but the same issue is present in Pollux, albeit with fewer possible interpretations of an analyst interaction. Thus, the introduction of clusters simplifies but does not solve "With Respect to What."

4.8 Multiple Distance Functions and Weight Vectors

Our implementation of Pollux uses a single shared distance function and weight vector for the Dimension Reduction and Clustering Models. However, implementations could certainly be produced that learn separate weight vectors for each model, each of which could then use a difference distance function in processing the dataset. For example, the Dimension Reduction Model, using Manhattan distance to boost computational efficiency, might use a different weight vector than the Clustering Model, which still uses Euclidean distance to accurately determine distances between clusters in the high-dimensional data.

5 Evaluation

In this section, we evaluate Pollux via a case study, performing reclassification interactions on the Census dataset to create a particular cluster of states, and evaluate the attribute weights learned within the system to create such an overall clustering and layout. After normalizing this dataset, we create an initial clustered projection (right panel of Fig. 1) in which each of the 35 dimensions begins with a weight of 1. Processing this dataset with the *k*-means algorithm produces six clusters.

The Census Bureau defines the Midwest Region as a collection of 12 states, ranging from Ohio in the east to the Dakotas in the

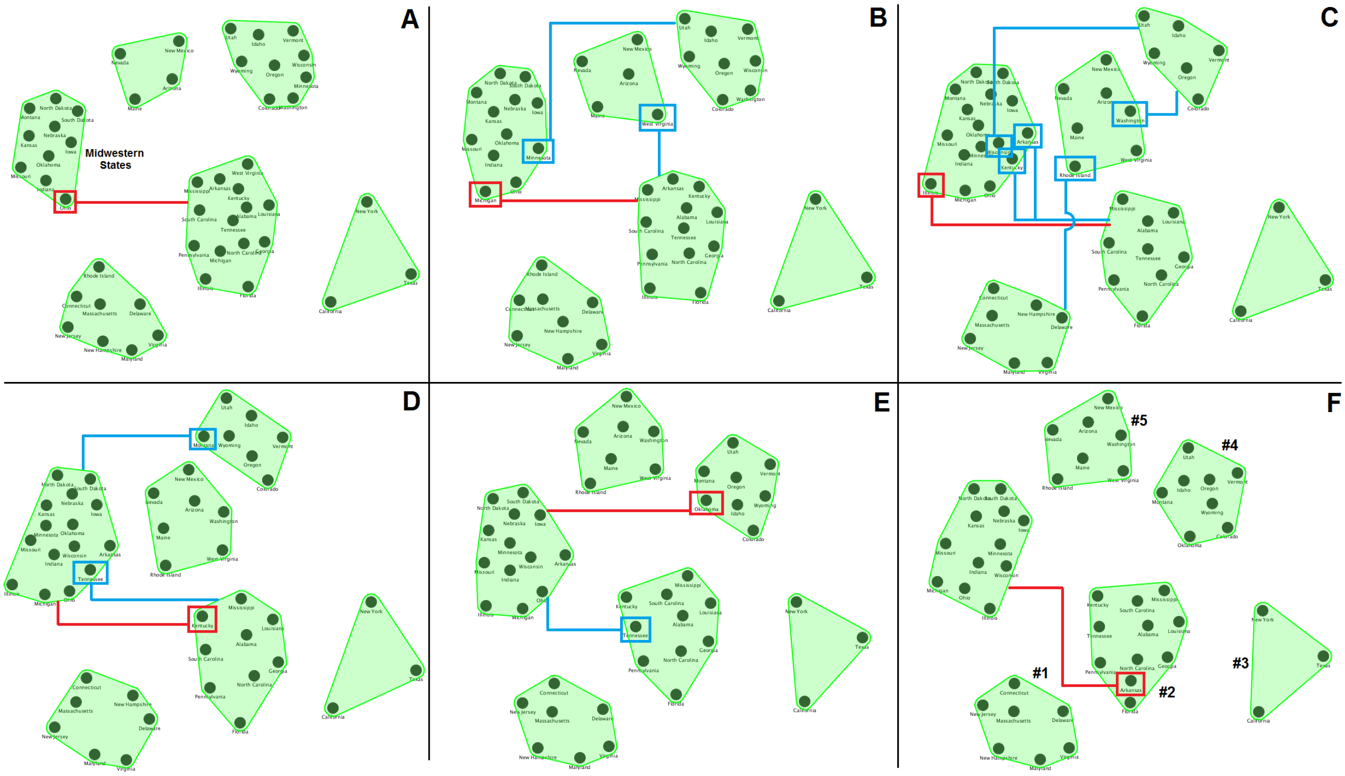

Figure 8: Each of the six interactions performed by the analyst in the case study. Nodes enclosed by red rectangles denote analyst-driven classification updates, while nodes enclosed by blue rectangles denote classification updates made by the system in response to newly-learned weights. Lines are drawn to show observation paths from source to destination cluster.

west [52]. In the initial projection, the cluster annotated with “Midwestern States” on the left side of Fig. 8A already incorporates 7 of these 12 states (as well as two extra states). In order to create a Midwest Region cluster, we perform the reclassification interactions listed in the following paragraphs. The learning routine executes after each of the interactions is performed, learning new weights to reflect what has been learned about the intent of the analyst thus far.

The first interaction reclassifies Ohio as a Midwestern state, dragging it from the central center into the Midwestern States cluster (Fig. 8A). The dimension weights are updated to reflect both the departure of Ohio from its source cluster and its introduction into the target cluster. Following the weight updates, no other states have received new cluster assignments. However, the area of the Midwestern States expands slightly, both because the number of nodes has increased and because Ohio is pushed to the outskirts of the cluster. The second occurs because the system has only recorded a single interaction; it has not yet learned enough to understand the optimal position of Ohio within the cluster.

The second interaction is similar, reclassifying Michigan as a Midwestern state by transferring it from the central cluster into the Midwestern States cluster (Fig. 8B). Following the weight updates, several changes are now apparent in both the clustering assignments and in the overall projection. Minnesota was pulled into the forming Midwestern States cluster from the cluster to the upper-right, while West Virginia was reclassified as a member of the top-center cluster, departing the central cluster. Further, the three upper clusters begin moving closer together as a result of ethnicity weights exerting more influence upon the overall graph. The states in these three clusters all have similar ethnic breakdowns, causing this effect.

The third interaction causes substantially more updates. Reclassifying Illinois as a Midwestern state also brings in Wisconsin (intended), as well as Arkansas and Kentucky (unintended) (Fig. 8C). As a result, there are now four states which must be removed from the cluster, but all five new states have now been introduced. In addition to these updates to the Midwestern States cluster, the states of Rhode Island and Washington were transferred into the upper-center cluster. The three upper clusters continue their drift from the previous interaction.

The fourth interaction begins the removal of the unwanted states, and also demonstrates the inertia of the learning routine. Removing Kentucky from the Midwestern States cluster and positioning it into the cluster which appears most sensible based on geography (the center cluster) also results in the automatic removal of Montana (hoped for), but it additional brings unwanted Tennessee into the Midwestern States cluster (Fig. 8D). Tennessee and Kentucky are quite similar states, and Tennessee was close to being relocated into the Midwestern States cluster before this interaction. Following the removal of Kentucky, the dimension weights just enough to finally pull Tennessee in. The upper-center and upper-right clusters were temporarily overlapping after this interaction, though they eventually separated as the projection stabilized.

Finally, the fifth and sixth interactions had minimal impact beyond the removal of states from the Midwestern States cluster. Reclassifying Oklahoma into the upper-right cluster removed Tennessee and returned it to the central cluster, where it was classified prior to the fourth interaction (Fig. 8E). Reclassifying Arkansas into the central cluster had no other cluster assignment effects (Fig. 8F).

The result of this set of six interactions is the formation of a cluster of the 12 Midwestern states, as well as five other clusters which saw occasional updates based upon the analyst’s interactions with the Midwestern States cluster. Progressing counterclockwise (with the clusters labeled #1–5 in Fig. 8F), these clusters could be

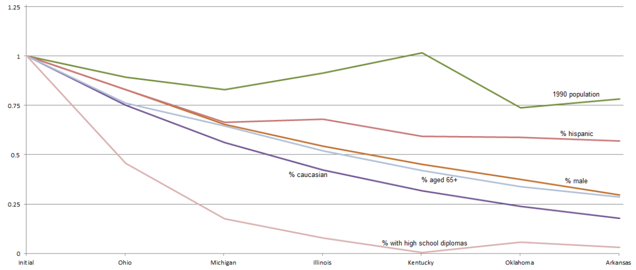

Figure 9: A selection of six attribute weights and their respective value updates during the six analyst interactions.

mapped with semantic meanings such as Northeastern States (#1), East Coast States (#2), High-Population States (#3), Low-Population States (#4), and a cluster that is difficult to label, but contains states that lean towards elderly, rural populations with lower than average per capita income (#5).

Fig. 9 shows a selection of six attribute weights, their value updates following each analyst interaction, and their influence on the overall projection as a result. The weight with consistently the most influence over the projection, the percentage of residents with a high school diploma, matches well with the states that the analyst reclassified: Ohio, Michigan, and Illinois each rate between #21 and #26 when the states are sorted by this attribute, while Kentucky, Oklahoma, and Arkansas are #2, #15, and #4 respectively. Indeed, the only time when the high school graduation rate decreases in influence was after Oklahoma was reclassified. The attribute weights associated with Caucasian, elderly, and male residents consistently declined through the interaction sequence, while the attribute weight for Hispanic residents varied in influence by that group's population in each state. The 1990 population was selected in the figure to demonstrate a dimension that seemed to have no meaningful impact on the overall projection, with the value of this attribute weight oscillating about the default of 1 as the interaction sequence progressed.

6 Discussion

The overarching focus of this research direction is to continue to explore the complex interplay between dimension reduction and clustering algorithms in both systems and in humans. As noted in the introduction, these algorithms serve different cognitive purposes but can naturally coexist within a projection of high-dimensional data. Understanding how analysts interpret and interact with such visualizations is a long-term goal, of which Pollux represents one point in the overall design space. That said, the variety of visual representations that can be produced by Pollux through modifications to the edge class selection and weights, cluster representation, distance function selection, and learning method demonstrate the flexibility of this framework for visualizing high-dimensional datasets.

6.1 Benefits of Subspace Clustering

The clustering assignments and projections in Pollux current make use of a global weight vector, but this does not necessarily need to be the case. Clusters are intended to group related observations, but when the number of dimensions becomes too great (as in the Animals and Census datasets), some dimensions may not be meaningful for a given cluster. It is further likely that some dimensions are correlated, and as such do not provide new information to the clustering process. The goal of subspace clustering is to identify a smaller, relevant set of dimensions that can be used to structure a particular cluster [36]. Of particular note to visualization research is biclustering [31, 42], which approaches the subspace clustering problem by striving to simultaneously cluster both observations and dimensions in order to identify pockets of similar behavior within a larger dataset. These subspace techniques offer an alternative method to creating and spatializing clusters.

6.2 Limitations and Future Work

One notable limitation of the work presented here is the scale of data tested. Though we did visualize datasets of various shapes (e.g., many more observations than dimensions, similar numbers of observations and dimensions), the true power of this cluster-first technique lies in its scalability. The current *k*-means and force-directed implementation of Pollux was the limiting factor in experimenting with larger data scale, and so we chose to focus this work on demonstrating the use and success of the reclassification, learning, and layout technique. Our next step in development is to re-implement the system with scalability in mind.

Additionally, our focus in this work to date has been exploring the visualization space of cluster-first projections specifically. When introducing this into a more complete application, the existing interface can be supplemented with a variety of additional views. These views can incorporate visualization of attribute weights, both at the current time (as seen in Andromeda [47]) and over the span of many interactions (as shown in Fig. 9). The quality of the clustering across an interaction set can also be visualized, making use of a metric such as the Dunn Index [20], the Davies-Bouldin Index [13], or the Silhouette Coefficient [45]. Such Explainable AI techniques provide the analyst with additional insight beyond the projection and clustering assignments themselves.

Further, we have not yet performed a user study to test the usability of this cluster-first technique in contrast to existing layout-first techniques. Our demonstration of Pollux in this work is limitation to a case study demonstration of the technique. A full study to examine the similarities and differences in insights generated by each technique is currently planned.

Finally, a limitation of the Pollux technique is the distortion of space to create compact clusters, a distortion that goes beyond that which is already necessary when projecting into a low-dimensional space. Beyond examining the underlying model weights, Pollux currently lacks a method for demonstrating to analysts whether or not their current clustering is meaningful. In other words, is the clustering that was constructed by the analyst supported by the data, or does it force nonsensical constraints upon the data in order to generate the current clustering? There are a variety of methods that we are considering to visualize the quality of clusters and to quantify the fitness of user-imposed constraints, with enough options under discussion to necessitate a further study, taking this issue beyond the scope of this work.

7 Conclusion

This work presents Pollux, a system that combines clustering and dimension reduction algorithms in a cluster-first framework to efficiently produce an interactive visualization of an input dataset. By interacting with the visualization, an analyst provides feedback to the underlying models, incrementally training the models to produce representations that reflect the current exploration interests of the analyst. We discuss means by which the default implementation of Pollux can be altered or extended, and we demonstrate the effectiveness of the interactive reclassification interaction via a case study. The flexibility demonstrated by Pollux presents an interesting tool to continue to develop, with several future studies planned.

Acknowledgments

Many thanks are given to the members of the Virginia Tech Bayesian Visual Analytics and InfoVis research groups for feedback during the development of this work.

References

[1] J. Abello, F. V. Ham, and N. Krishnan. Ask-graphview: A large scale graph visualization system. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):669–676, Sept 2006. doi: 10.1109/TVCG.2006.120

[2] G. Andrienko, N. Andrienko, S. Rinzivillo, M. Nanni, and D. Pedreschi. A visual analytics toolkit for cluster-based classification of mobility data. In N. Mamoulis, T. Seidl, T. B. Pedersen, K. Torp, and I. Assent, eds., *Advances in Spatial and Temporal Databases*, pp. 432–435. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[3] G. Andrienko, N. Andrienko, S. Rinzivillo, M. Nanni, D. Pedreschi, and F. Giannotti. Interactive visual clustering of large collections of trajectories. In *2009 IEEE Symposium on Visual Analytics Science and Technology*, pp. 3–10, Oct 2009. doi: 10.1109/VAST.2009.5332584

[4] J. Barnes and P. Hut. A hierarchical o (n log n) force-calculation algorithm. *Nature*, 324(6096):446, 1986.

[5] S. Basu, D. Fisher, S. M. Drucker, and H. Lu. Assisting users with clustering tasks by combining metric learning and classification. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.

[6] L. Boudjeloud-Assala, P. Pinheiro, A. Blansch, T. Tamisier, and B. Otjacques. Interactive and iterative visual clustering. *Information Visualization*, 15(3):181–197, 2016. doi: 10.1177/1473871615571951

[7] 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)*, pp. 163–172, Oct 2014. doi: 10.1109/VAST.2014.7042492

[8] 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)*, pp. 83–92, Oct 2012. doi: 10.1109/VAST.2012.6400486

[9] E. T. Brown, S. Yarlagadda, K. Cook, R. Chang, and A. Endert. Modelspace: Visualizing the trails of data models in visual analytics systems. In *Proceedings of the Machine Learning from User Interaction for Visualization and Analytics Workshop at IEEE VIS 2018*, Oct 2018.

[10] J. Choo, C. Lee, C. K. Reddy, and H. Park. Utopian: User-driven topic modeling based on interactive nonnegative matrix factorization. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):1992–2001, Dec 2013. doi: 10.1109/TVCG.2013.212

[11] 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.

[12] A. Coden, M. Danilevsky, D. Gruhl, L. Kato, and M. Nagarajan. A method to accelerate human in the loop clustering. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pp. 237–245. SIAM, 2017.

[13] D. L. Davies and D. W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227, April 1979. doi: 10.1109/TPAMI.1979.4766909

[14] C. Ding and T. Li. Adaptive dimension reduction using discriminant analysis and k-means clustering. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pp. 521–528. ACM, New York, NY, USA, 2007. doi: 10.1145/1273496.1273562

[15] V. Dobrynin, D. Patterson, M. Galushka, and N. Rooney. Sophia: an interactive cluster-based retrieval system for the ohsumed collection. *IEEE Transactions on Information Technology in Biomedicine*, 9(2):256–265, June 2005. doi: 10.1109/TITB.2005.847184

[16] E. P. dos Santos Amorim, E. V. Brazil, J. Daniels, P. Joia, L. G. Nonato, and M. C. Sousa. ilamp: Exploring high-dimensional spacing through backward multidimensional projection. In *2012 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 53–62, Oct 2012. doi: 10.1109/VAST.2012.6400489

[17] M. Dowling, N. Wycoff, B. Mayer, J. Wenskovitch, S. Leman, L. House, N. Polys, C. North, and P. Hauck. Interative visual analytics for sensemaking with big text. *Journal of Big Data Research, Special Issue on Big Data Exploration, Visualization, & Analytics*, 2019.

[18] D. Dua and C. Graff. UCI machine learning repository, 2017.

[19] A. Dubey, I. Bhattacharya, and S. Godbole. A cluster-level semi-supervision model for interactive clustering. In J. L. Balcázar, F. Bonchi, A. Gionis, and M. Sebag, eds., *Machine Learning and Knowledge Discovery in Databases*, pp. 409–424. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[20] J. C. Dunn. Well-separated clusters and optimal fuzzy partitions. *Journal of cybernetics*, 4(1):95–104, 1974.

[21] M. El-Assady, R. Sevastjanova, F. Sperrle, D. Keim, and C. Collins. Progressive learning of topic modeling parameters: A visual analytics framework. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):382–391, Jan 2018. doi: 10.1109/TVCG.2017.2745080

[22] A. Endert, L. Bradel, and C. North. Beyond control panels: Direct manipulation for visual analytics. *IEEE Computer Graphics and Applications*, 33(4):6–13, July 2013. doi: 10.1109/MCG.2013.53

[23] A. Endert, R. Chang, C. North, and M. Zhou. Semantic interaction: Coupling cognition and computation through usable interactive analytics. *IEEE Computer Graphics and Applications*, 35(4):94–99, July 2015. doi: 10.1109/MCG.2015.91

[24] A. Endert, P. Fiaux, and C. North. Semantic interaction for sensemaking: Inferring analytical reasoning for model steering. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2879–2888, Dec 2012. doi: 10.1109/TVCG.2012.260

[25] A. Endert, P. Fiaux, and C. North. Semantic interaction for visual text analytics. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pp. 473–482. ACM, New York, NY, USA, 2012. doi: 10.1145/2207676.2207741

[26] A. Endert, M. S. Hossain, N. Ramakrishnan, C. North, P. Fiaux, and C. Andrews. The human is the loop: new directions for visual analytics. *Journal of Intelligent Information Systems*, 43(3):411–435, 2014. doi: 10.1007/s10844-014-0304-9

[27] V. Estivill-Castro. Why so many clustering algorithms: A position paper. *SIGKDD Explor. Newsl.*, 4(1):65–75, June 2002. doi: 10.1145/568574.568575

[28] J. C. Gower. A general coefficient of similarity and some of its properties. *Biometrics*, 27:857–871, 1971.

[29] R. L. Graham and F. F. Yao. Finding the convex hull of a simple polygon. *Journal of Algorithms*, 4(4):324–331, 1983.

[30] P. Guo, H. Xiao, Z. Wang, and X. Yuan. Interactive local clustering operations for high dimensional data in parallel coordinates. In *2010 IEEE Pacific Visualization Symposium (PacificVis)*, pp. 97–104, March 2010. doi: 10.1109/PACIFICVIS.2010.5429608

[31] J. A. Hartigan. Direct clustering of a data matrix. *Journal of the American Statistical Association*, 67(337):123–129, 1972. doi: 10.1080/01621459.1972.10481214

[32] E. Hoque and G. Carenini. Convisit: Interactive topic modeling for exploring asynchronous online conversations. In *Proceedings of the 20th International Conference on Intelligent User Interfaces*, IUI '15, pp. 169–180. ACM, New York, NY, USA, 2015. doi: 10.1145/2678025.2701370

[33] Y. Hu, J. Boyd-Graber, B. Satinoff, and A. Smith. Interactive topic modeling. *Machine Learning*, 95(3):423–469, Jun 2014. doi: 10.1007/s10994-013-5413-0

[34] P. Joia, D. Coimbra, J. A. Cuminato, F. V. Paulovich, and L. G. Nonato. Local affine multidimensional projection. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2563–2571, Dec 2011. doi: 10.1109/TVCG.2011.220

[35] M. Kim, K. Kang, D. Park, J. Choo, and N. Elmqvist. Topiclens: Efficient multi-level visual topic exploration of large-scale document collections. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):151–160, Jan 2017. doi: 10.1109/TVCG.2016.2598445

[36] H.-P. Kriegel, P. Kröger, and A. Zimek. Subspace clustering. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(4):351–364, 2012.

[37] C. H. Lampert, H. Nickisch, S. Harmeling, and J. Weidmann. Animals with attributes: A dataset for attribute based classification, 2009.

[38] J. A. Lee and M. Verleysen. *Nonlinear dimensionality reduction*. Springer Science & Business Media, 2007.

[39] S. C. Leman, L. House, D. Maiti, A. Endert, and C. North. Visual to parametric interaction (V2PI). *PloS one*, 8(3):e50474, 2013.

[40] J. MacInnes, S. Santosa, and W. Wright. Visual classification: Expert knowledge guides machine learning. *IEEE Computer Graphics and Applications*, 30(1):8–14, Jan 2010. doi: 10.1109/MCG.2010.18

[41] G. M. H. Mamani, F. M. Fatore, L. G. Nonato, and F. V. Paulovich. User-driven feature space transformation. *Computer Graphics Forum*, 32(3pt3):291–299, 2013. doi: 10.1111/cgf.12116

[42] B. Mirkin. *Mathematical classification and clustering*. Kluwer Academic Publishers, 1996.

[43] V. Molchanov and L. Linsen. Interactive Design of Multidimensional Data Projection Layout. In N. Elmqvist, M. Hlawitschka, and J. Kennedy, eds., *EuroVis - Short Papers*. The Eurographics Association, 2014. doi: 10.2312/eurovisshort.20141152

[44] F. Paulovich, D. Eler, J. Poco, C. Botha, R. Minghim, and L. Nonato. Piecewise laplacian-based projection for interactive data exploration and organization. *Computer Graphics Forum*, 30(3):1091–1100, 2011. doi: 10.1111/j.1467-8659.2011.01958.x

[45] P. J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[46] J. Z. Self, M. Dowling, J. Wenskovitch, I. Crandell, M. Wang, L. House, S. Leman, and C. North. Observation-level and parametric interaction for high-dimensional data. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 8(2):15:1–15:36, 2018. doi: 10.1145/3158230

[47] 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, pp. 3:1–3:6. ACM, New York, NY, USA, 2016. doi: 10.1145/2939502.2939505

[48] F. M. Shipman III and R. McCall. Supporting knowledge-base evolution with incremental formalization. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 285–291, 1994.

[49] A. Singhal. Modern information retrieval: A brief overview. *IEEE Data Eng. Bull.*, 24(4):35–43, 2001.

[50] O. Sourina and D. Liu. Visual interactive clustering and querying of spatio-temporal data. In O. Gervasi, M. L. Gavrilova, V. Kumar, A. Laganá, H. P. Lee, Y. Mun, D. Taniar, and C. J. K. Tan, eds., *Computational Science and Its Applications – ICCSA 2005*, pp. 968–977. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[51] R. L. Thorndike. Who belongs in the family? *Psychometrika*, 18(4):267–276, 1953.

[52] United States Census Bureau. Census Regions and Divisions of the United States, 2016.

[53] United States Census Bureau. State Data Center Program, 2019.

[54] J. Wenskovitch, L. Bradel, M. Dowling, L. House, and C. North. The effect of semantic interaction on foraging in text analysis. In *2018 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 13–24, Oct 2018. doi: 10.1109/VAST.2018.8802424

[55] J. Wenskovitch, I. Crandell, N. Ramakrishnan, L. House, S. Leman, and C. North. Towards a systematic combination of dimension reduction and clustering in visual analytics. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):131–141, Jan 2018. doi: 10.1109/TVCG.2017.2745258

[56] J. Wenskovitch and C. North. Observation-level interaction with clustering and dimension reduction algorithms. In *Proceedings of the 2nd Workshop on Human-In-the-Loop Data Analytics*, HILDA'17, pp. 14:1–14:6. ACM, New York, NY, USA, 2017. doi: 10.1145/3077257.3077259