Bridging the Gap between User Intention and Model Parameters for Human-in-the-Loop Data Analytics

Jessica Zeitz Self, Radha Krishnan Vinayagam, J.T. Fry, Chris North
Discovery Analytics Center, Virginia Tech
Blacksburg, VA
{ jzself, vradha, fryjt1, north } @vt.edu

ABSTRACT

Exploratory data analysis is challenging given the complexity of data. Models find structure in the data lessening the complexity for users. These models have parameters that can be adjusted to explore the data from many different angles providing more ways to learn about the data. "Human in the loop" means users can interact with the parameters to explore alternative structures. This exploration allows for discovery. This paper examines usability issues of Human-Model Interaction (HMI) for data analytics. In particular, we bridge the gaps between a user's intention and the parameters of a WMDS model during HMI communication.

CCS Concepts

• **Human-centered computing** → Human computer interaction (HCI) → HCI design and evaluation methods → User studies.

Keywords

Visual analytics; object-level interaction; usability.

1. INTRODUCTION

Mathematical models are only useful to a user if they can correctly communicate with them. Learning how to use models can be difficult because of their complexity. It is particularly problematic for non-experts who are not as familiar with the details of the model and the meanings of the parameters. However, supplying model parameters and interpreting model feedback is a challenging task for a non-expert of the model. This challenge presents an interaction design problem. This paper's contributions are as follows:

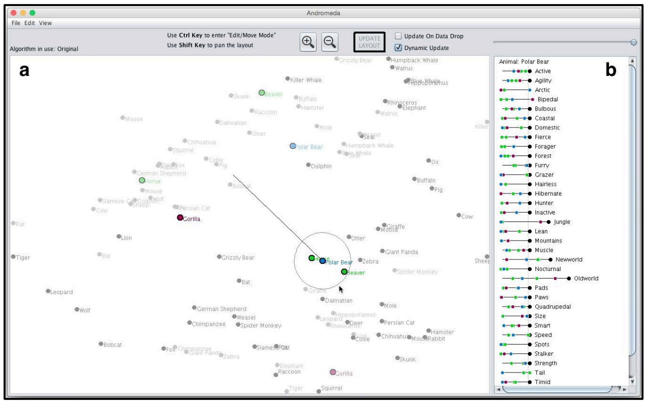
1. Explore these issues in context of Weighted Multidimensional Scaling (WMDS) for high-dimensional data analytics.
2. Identify usability problems in interacting with WMDS.
3. Create and evaluate interaction design solutions for these problems.

The interaction design solutions seek to enhance Human-Model Interaction (HMI). Our research abstracts the model parameters and transforms them into the space of a visual metaphor. Users interact with the visual space while indirectly adjusting the parameters. Object-level interaction (OLI) provides an intuitive means to communicate with a model to analyze data. We have designed interactions that bridge the intent of the cognitive model to the parameters of the math model, specifically Weighted Multidimensional Scaling (WMDS). We emphasize the visual metaphor of mapping similarity to proximity, which is intuitive for non-expert users. Despite the metaphor familiarity, there are still disconnects between user intention and the model parameters. This paper discusses issues users had and the designed interaction solutions and algorithmic modifications needed to alleviate these issues. We present a usability study that tested the new interaction designs.

2. OBJECT-LEVEL INTERACTION

When analyzing data, users want to ask questions about and find similarities between the data points. With parametric interaction (PI), users have to answer these questions indirectly by trial and error adjusting individual parameters. This is a HMI gap. Object-level interaction provides a solution [2]. With OLI, users directly manipulate the data points injecting their own knowledge and questions. This requires a redesign of the dimension reduction algorithm to calculate the reverse optimization.

Object-level interaction (OLI) happens within a visualization displaying a projection generated by a dimension reduction algorithm. Specifically, Weighted Multidimensional Scaling (WMDS) [6] reduces a dataset from high-dimensional space to low-dimensional space while mostly preserving the high-dimensional pairwise distances in the low-dimensional projection. WMDS takes a weight for each dimension representing importance to the user as input parameters. The model outputs a visualization of objects in low-dimensional space (i.e., 2D) where the relative distance between objects represents similarity. A user


Algorithm in use: Original

Use **Ctrl Key** to enter "Edit/Move Mode"
Use **Shift Key** to pan the layout

□ Update On Data Drop
☑ Dynamic Update

Animal: Polar Bear

Figure 1. Screenshot of Andromeda interface. (a) The object view visualizes the 2D projection (dark gray points) calculated by WMDS. Ghost preview points (light gray) are visible based on the user moving the Polar Bear (blue) away from the Gorilla (maroon) and toward the Horse and Beaver (green). (b) The parameter view displays the weights associated with the object visualization.

HILDA'16, June 26, 2016, San Francisco, CA, USA.

ACM 978-1-4503-4207-0/16/06 $15.00.
DOI: http://dx.doi.org/10.1145/2939502.2939505

can directly manipulate the objects within the visual space instead of adjusting the weight parameters. OLI provides a means of communication between a user and a model that bridges the gap.

To study OLI, we designed and developed an interactive interface, Andromeda (Figure 1), that visualizes high-dimensional data using WMDS [8]. Andromeda's object view (Figure 1a) visualizes the WMDS projection and the parameter view (Figure 1b) displays the weights as horizontal lines. For parametric interaction, a user adjusts the weights by dragging the handle on the line to the right or left to increase or decrease the weight respectively. A user can perform OLI by dragging the objects and the system recalculates new weights to explain the new positions of the objects [4, 5, 7]. This paper focuses on the specific interactions we designed to support communication between a user and the WMDS model.

Previous research involving a controlled user study found the interactions within Andromeda allowed users to perform successful data analyses [8]. OLI and parametric interaction provide the user with two analysis angles: object-centric and dimension-centric. OLI enabled new insights about clustering and outliers which focus on relationships between objects.

Despite the positive results of the study, usability issues arose proving there are still disconnects between user intentions when performing and perceiving the interactions and the model parameters. When performing an interaction, users are injecting information to the model through the system. The system has to glean model parameters from the interactions.

One problem already encountered was that users intent was to express relationships between a subset of the points and the users were not interested in all of the points in the dataset. However, the algorithm assumed all points were intended. This is a mismatch between the cognitive model of the user and mathematical model. Thus, Hu et al. [5] introduced the idea of highlighting where the user specifies all points that are relevant to the interaction and the algorithm only calculates based on this subset of points.

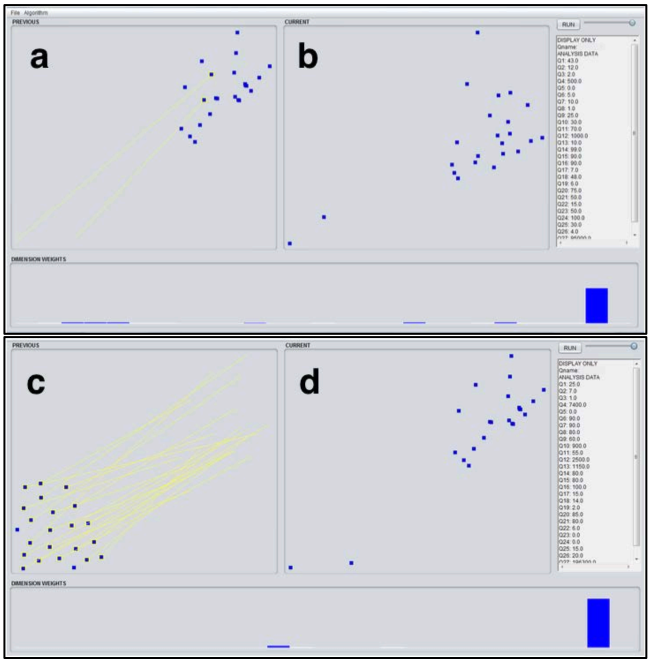
This solution produced yet more HMI problems. In this paper, we outline these usability issues and provide new design choices we developed to combat the issues.

3. USABILITY PROBLEMS, SOLUTIONS, RESULTS

To test the solutions, we performed a quasi-empirical usability study [3]. Seven graduate students explored a high-dimensional dataset about 49 animals and 36 dimensions. Most participants had never heard of Multidimensional Scaling and no participant had ever used it. Each participant answered a series of questions aimed at answering whether the usability problems had been rectified with our new interactions. Participants also rated Andromeda using the System Usability Scale (SUS) [1].

3.1 With Respect to What?

Problem. When dragging points around within a visualization, a user probably has a goal in mind. A common goal we have seen during analyses of high-dimensional data is to figure out how outlier(s) fit into a cluster of points. To achieve this answer, users drag the outlier into the cluster of points and click to update the algorithm. Figure 2a shows the original Andromeda interface where a user tried to drag two outliers into a cluster. The resulting layout (Figure 2b) shows the outliers in their original location because the relative distances between the two user-moved points (the outliers) did not change.



DISPLAY ONLY
Qname:
ANALYSIS DATA
Q1: 43.0
Q2: 12.0
Q3: 2.0
Q4: 500.0
Q5: 0.0
Q6: 5.0
Q7: 10.0
Q8: 1.0
Q9: 25.0
Q10: 30.0
Q11: 70.0
Q12: 1000.0
Q13: 10.0
Q14: 99.0
Q15: 90.0
Q16: 90.0
Q17: 7.0
Q18: 48.0
Q19: 6.0
Q20: 75.0
Q21: 50.0
Q22: 15.0
Q23: 50.0
Q24: 100.0
Q25: 30.0
Q26: 4.0

DISPLAY ONLY
Qname:
ANALYSIS DATA
Q1: 25.0
Q2: 7.0
Q3: 1.0
Q4: 7400.0
Q5: 0.0
Q6: 90.0
Q7: 90.0
Q8: 80.0
Q9: 60.0
Q10: 900.0
Q11: 55.0
Q12: 2500.0
Q13: 1150.0
Q14: 80.0
Q15: 80.0
Q16: 100.0
Q17: 15.0
Q18: 14.0
Q19: 2.0
Q20: 85.0
Q21: 80.0
Q22: 6.0
Q23: 0.0
Q24: 0.0
Q25: 15.0
Q26: 20.0

Figure 2. Andromeda interface depicting usability issue. (a) User drags two outliers into cluster to find out what makes the outliers similar to the cluster. (b) Result of interaction where outliers return to original location since no points in cluster were moved. (c) User tries opposite interaction and drags all data points in cluster toward outliers. (d) Same result as before where cluster returns to original location.

The particular user tried the next best option and moved the entire cluster of points toward the outliers. The resulting layout shows the same phenomenon where the cluster returns to its original location away from the outliers. The user moved the outliers with respect to the cluster of points; however, the algorithm did not receive that input. The interaction did not work because it is important for the algorithm to consider what points the user was dragging the outliers away from and what points she was dragging the outliers toward.

The algorithm only considers points that have been explicitly highlighted by the user to increase efficiency and speed; therefore, in this scenario the algorithm only considers the outliers and not the points in the cluster. This mismatch causes a usability problem where the user is assuming the points near to the moved point will be considered, however this is not the case. We had to design a visual cue to help the user recognize the need to be explicit to help the algorithm know what points are important. The visual cue clarifies the communication between the user and model.

Solution. To resolve this problem, the interface automatically highlights reference points near the point being moved by the user; specifically, points being moved away from and points being moved towards. We implemented a radius that surrounds the data point the user is dragging (Figure 3). When a user clicks to move a data point, all points within the radius of this original location are automatically highlighted (maroon points) and considered by the algorithm. Once the user drops the data point she is moving, radial (green) points within this new location are also automatically highlighted and considered by the algorithm. The interface is automatically eliciting algorithm input from the user which improves the result. These automatic highlights also satisfy the need to move at least three points to specify a change in

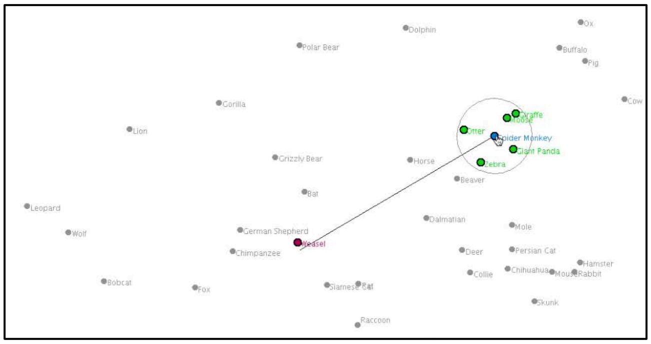


Figure 3. Visualization of user interaction dragging the Spider Monkey away from the maroon data point and toward the cluster of green data points. Andromeda automatically highlights the maroon and green points to distinguish the data points the user is dragging away from and toward.

relative distances. Away from and towards points make sure the relative distances change and visually notate to the user which points are away from (maroon) and which are toward (green). If there are no points within the radius, the closest point is highlighted. It is a visual cue for users that they can highlight other points.

Table 1. Number of Participants who Performed Each Type of Interaction to Answer Outlier Questions

	Q1	Q2	Q3
Viewed raw data	3	2	2
OLI	4	1	1
PI	0	0	0
Raw data & OLI	0	2	2
Raw data & PI	0	2	2

Results. To test this new interaction, we asked users to find similarities between an outlier and a cluster of points far from the outlier. Users answered two such questions: (Q1) "How is the killer whale similar to the cluster with the German shepherd, chimpanzee, spider monkey, and weasel" and (Q2) "How is the tiger similar to the cluster with the blue whale, elephant, hippopotamus, and rhinoceros". Users answered both questions using one of two interactions. Table 1 shows how many users performed each interaction for the two questions. Users either selected the data points in the visualization and browsed the raw data on the weight lines looking for tight clusters depicting similar values or users performed object-level interaction dragging the outlier toward the cluster and looking at the upweighted dimensions. Four participants used a combination of these interactions along with parametric interaction. Some users manually unhighlighted automatically highlighted points indicating that they understand this visual cue.

3.2 Creating a New Cluster

Problem. Users often want to make a new cluster out of disparate points by dragging them together into an empty area. However, they typically do not think to specify the points they are moving away from. This is a communication problem because inverse WMDS requires changes in relative distances, meaning that some points must move closer while other points move further apart. If the relative distances between pairs of points in the visualization has not changed, then the weight vector calculated by the optimization will also not be changed. Imagine there is a cluster of three points that are all equidistant from each other in the visualization (see Figure 4a). A user then drags each of these three points even closer together. The three points are still equidistant to each other despite the actual distance being smaller; the ratio of the pairwise distances has remained the same (see Figure 4b).

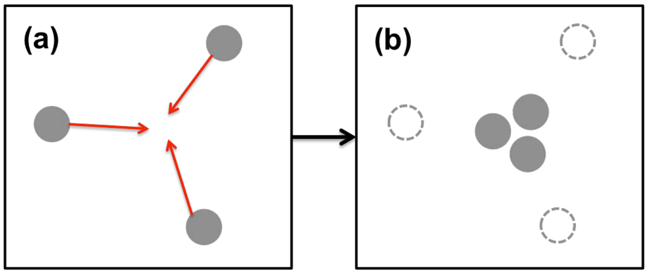


Figure 4. This figure visualizes how dragging three points does not change the relative distances between the points. (a) Three points are visualized and the red arrows indicate a user moving the points closer within the space. (b) The new visualization displays the three points closer to each other, however the relative distances have not changed from (a).

Understanding the relative distances is challenging for users especially since the points within the visualization have actual distances. It looks like the user has modified the points enough to get new information from the algorithm if the user is thinking about actual distance. The algorithm calculating on relative distances does not match the user's expectation which is actual distances. The visualization and interactions must convey the notion of relative distances to the user.

Solution. The radius we designed helps alleviate this issue. By automatically highlighting other points that are near the original location of the points being moved, the relative distances are changing. The more points the algorithm considers, the less probability for unchanged relative distances. However, there are some cases that are not solved by this solution. If users manually unhighlight the nearby points, the solution fails.

A further solution involves modifying the model to consider the relative distances between the moved points with respect to all other points. The current WMDS scheme first finds weights based on user OLI and then projects the high dimensional coordinates into 2-dimensions using the calculated weights. An improved optimization will perform both steps simultaneously by utilizing several constraints that will alleviate the issue with unchanged relative distances. Consider the following equations (1)-(4).

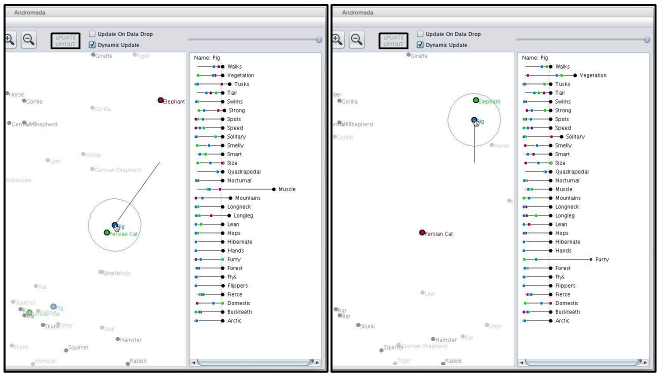
$$\underset{w_1,\ldots,w_p,\ldots\mathbf{z}_1,\ldots,\mathbf{z}_n}{\text{ArgMin}} \sum_{i<j} \left(\|\mathbf{x}_i - \mathbf{x}_j\|_w - \|\mathbf{z}_i - \mathbf{z}_j\| \right)^2 \quad (1)$$

$$\text{Subject to} \sum_{i<j} \|\mathbf{z}_i - \mathbf{z}_j\| = 1 \quad (2)$$

$$\sum_{i<j} \|\mathbf{z}_i^* - \mathbf{z}_j^*\| = \pi \quad (3)$$

$$\sum_{i=1}^{p} w_i = 1 \quad (4)$$

We will let **x** denote a high dimensional point, **z** a 2-dimensional point, $\mathbf{z}^*$ a 2-dimensional point that the user has moved, and w an attribute weight. Equations (1) and (4) are of the familiar form used in the current WMDS optimization, except that we will find weights and low dimensional coordinates together. Equations (2) and (3), however, will account for interactions with unchanged relative distances. Each projection will always be such that the sum of all pairwise distances between $\mathbf{z}_i$ and $\mathbf{z}_j$ is 1. This provides an upper bound for the scale of the projection. After the user interacts with points $\mathbf{z}_i^*$, we will calculate π - the ratio of the *sum of pairwise distances between the moved points* to the *sum of*


☐ Update On Data Drop
☑ Dynamic Update

Name: Pig

Figure 5. Two screenshots as a user moves the pig (blue) between the elephant and the Persian cat. Notice the weight changes based on the relative distances between the three animals and the relative distances of the raw data points on the emphasized weight lines.

pairwise distances for all points. If this ratio is small, it implies that the moved points should be close in comparison to the overall layout. Now, assume user drags points together such that the relative distances between each pair is unchanged. The algorithm now knows that the sum of distances between these points should be small relative to the sum of all pairwise distances. Enforcing this constraint for the updated visualization will preserve the information about OLI even with unchanged relative distances.

Results. To answer whether this issue has been fixed, we asked users to find similarities between three data points, the beaver, German shepherd and polar bear. These three points are equidistant from each other in the initial layout. If you drag the three together without considering other data points, then the relative distances do not change.

With the new interaction, users did not experience the original issue. As they dragged the three data points together, other points were automatically considered and the relative distances between the three data points moved by the user and the automatically chosen data points did change. All participants correctly identified the similar dimensions between the three data points.

3.3 User Confusion after Batch Update

Problem. The algorithm and optimization update in batch mode. Given a set of high-dimensional points and a weight vector, a dimension reduction algorithm returns a set of low-dimensional coordinates. Within the Andromeda interface, this update seems to happen abruptly to the user. A user increases the weight of one dimension and the visualization automatically updates the coordinates of the points without any transition. The same scenario occurs with OLI when a user modifies point locations and then clicks the run button to update the visualization. This may move the points to disparate locations, leaving the user confused and uncertain about what happened.

Solution. To solve this issue, we implemented an animation that transitions from the current projection to the new projection. After the algorithm calculates new coordinate locations, each data point smoothly transitions from its current location to its new location within the visualization. Users can repeat this animation in slow motion using a slider provided within the interface. This short animation gives a visual representation to the user of how the projection has changed given the updated calculations. A future enhancement is to interpolate the weights and animate through a series of projections on those weights.

Results. To test this feature, we asked the participants a series of guided questions which required the participant to explain the relative distances of particular data points in a new layout and why the data points transitioned to new coordinate locations. We asked participants to drag the dolphin close to a cluster that included the mouse, rabbit and hamster. All seven participants correctly listed the dimensions that increased in weight. Five participants correctly explained why the four data points relocated far from the seal; the cluster including the dolphin, mouse, rabbit and hamster is far from the seal because the clustered animals are similar based on activity, new world, and spots whereas the seal is very different. Six participants correctly identified why another animal of their choosing was placed far from the cluster. For example, one participant explained, "I can see the difference and similarities very clearly on the right panel. [The] Zebra is a grazer, bigger in size. All 5 are pretty closely active, foragers, timid therefore 0 [in] fierce, don't live in water. Zebra is not a new world animal. It is old world." The participants recognized that the layout depends on the weights of all dimensions. Emphasized dimensions have more effect on the coordinate locations of the data points.

3.4 Dynamic Exploration

Problem. The original Andromeda interface provided a user with parametric and object-level interactions. By dragging individual weight lines, the user can visualize how increasing and decreasing weights affect the projection of the high-dimensional data. Once the user released the mouse, the algorithm would update the projection. To add fluidity to a user's analysis, we modified this interaction and the algorithm dynamically updates the projection as the user is dragging the weight line. The user can explore increasing and decreasing one weight in a fluid motion while the algorithm continuously updates the projection.

After performing object-level interaction, the user clicks a button to start the update. On this button click, the algorithm takes the user input and updates the projection. The user bears the burden of deciding when the algorithm should perform the update. Having to make this decision interrupts the exploratory nature of analysis. During an analysis with this original button interaction, we noticed users tended to try object-level interaction once or twice and if it did not produce an expected result, they would completely switch inquiries instead of exploring further.

Solution. In order to provide an inviting exploratory environment, we designed dynamic updating for OLI and visual ghosted points. As a user drags one data point around the visualization, the algorithm is constantly updating the projection. Ghosted points appear to faintly visualize the new locations of all the points if the user were to drop the data point currently being dragged (see Figure 1). The user drags the polar bear (blue) away from the gorilla (maroon) and toward the horse and beaver (green). The line depicts the original location of the polar bear. The ghost points visualize the new locations of the data points if the user were to release the polar bear in that location. The weights are updated to reflect the projection of the ghost points. With dynamic updating, the user can continue to drag the polar bear around the visualization and get a preview of the projection and new weight vector. While a user is performing an analysis, it is inefficient and interrupts cognitive processes for him to wait for the model updates. Dynamic interaction allows for a more fluid exploratory process. Users interact with the progression of change. With dynamic interaction, the interface follows Shneiderman's direct manipulation principles [9]: the algorithm updates based on small incremental user adjustments which allows for rapid and reversible interactions and updates. For example, Figure 5 shows two previews as a user moves the pig between the elephant and Persian cat. The user sees a range of weights chosen based on the relative distances between these three animals.

Table 2. Interaction Counts Per Participant During Open-ended Analysis

Participant #	**Duration (min)**	**OLI**			**PI**	**Single Data Points Viewed**	**Cluster of Points Viewed**	**Points Hovered**
		Dynamic Update	*Automatic Update*	*Button Update*				
1	15.0	1	-	-	10	2	-	42
2	11.5	5	-	-	8	14	-	13
3	11.3	2	-	-	7	2	-	42
4	10.7	-	-	3	1	1	-	37
5	26.3	7	5	-	9	12	4	73
6	12.6	-	-	-	6	1	5	11
7	9.0	-	1	-	2	1	8	60

Results. During the open-ended analysis question, we asked participants to use any features of the interface to explore and learn about the data writing down insights you gain and appropriate rationale and evidence to back up your claims.

Table 2 shows the number of times each participant performed an interaction throughout his or her analysis. All but one participant performed some type of object-level interaction. Four participants utilized dynamic interaction. Two turned on automatic update so that when they moved one point, the layout automatically updated. One participant reverted back to the original interaction of clicking the button once he settled on a satisfying adjusted layout. All participants performed parametric interaction many times modifying many different dimensions.

3.5 What is the Rationale for the Layout?

Problem. The visualization and interactions we have designed are meant to ease the use of and hide the dimension reduction algorithm. By directly interacting with the projection and weights, users do not have to understand the details of the algorithm. However, for a more in depth analysis, a user must in some way understand why the visualization changed when the algorithm updates the projection and the weights. The user needs to understand the rationale behind the distances in the projection and that they are based on the weights. When a user modifies one (either the projection or the weights), what is the rationale for the change in the other? Why did the model place the points with certain distances? We provide visual cues and feedback to the user to help bridge this gap.

Solution. To strengthen the connection between the points in the projection and the dimensions, we visualize the raw data directly on the weight lines. When a point is selected within the projection, its raw data value is visualized as points on the weights lines. The location on the line depicts the numerical value; if the value is high, the point will appear toward the right and if the value is low, the point will appear closer to the left end of the line.

When a weight is increased, the similarities and differences between the data points are visually exaggerated since there is more emphasis on that particular weight. Users can see this exaggeration because the raw data of each point is drawn on the weight lines. When a data point is selected in the visualization, its raw data is drawn on all the dimension weight lines. In Figure 6a,

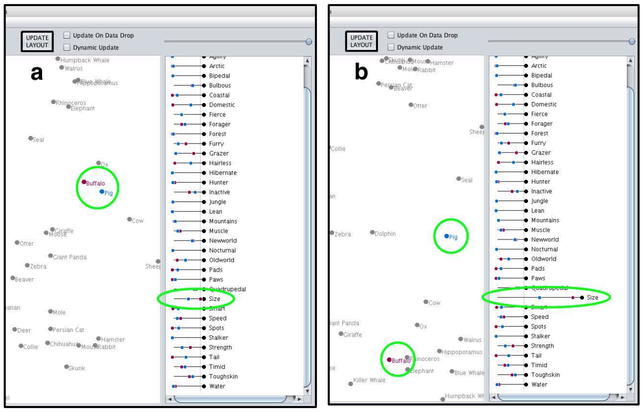


Figure 6. Two Andromeda screenshots depicting before and after a parametric interaction increasing the weight of the size dimension. (a) The Buffalo and Pig are close together in the projection, but have different raw data values for as seen on the size weight line. (b) After increasing the weight of the size dimension, the Buffalo and Pig are farther apart in the projection and the difference in raw data values is exaggerated on the size weight line since the user increased the importance of the size dimension.

Buffalo and Pig are selected and their corresponding raw data values are visualized on the weight lines. In particular, the Buffalo is bigger in size than the Pig since the maroon point is drawn toward the right end of the weight line whereas the blue point is toward the left. Figure 6b depicts a user increasing the importance of the size dimension. The blue and maroon points are now farther apart on the line because the difference between the sizes of the Buffalo and the Pig are amplified. Within the visualization, the Buffalo and Pig are farther apart since the user emphasized the size dimension where the Buffalo and Pig are different. The relative locations of the data points in the visualization is described by the relative locations of the raw data value data points on the weight lines.

Each visual cue we have designed further explains the dimension reduction process to the user. It is important for the user to understand the cause and effect of dragging a data point throughout the projection. If she drags one data point toward a cluster, what weights are increased and why those particular weights? Where do the data points move once the optimization is complete? To help the user understand the result, two groups of points are highlighted: data points being moved away from and data points being moved toward. Both sets of data points are notated in the projection as well as on the weight lines. As a user drags one single point, data points being dragged away from are colored maroon and data points being dragged toward are colored green. In the future, we want to design visual cues to strengthen the connection between the pairwise distances in the projection and the pairwise distances on the weight lines.

Results. In Figure 1, the user is dragging the *polar bear* away from the *gorilla* and toward the *horse* and *beaver*. We see the ghosted points within the projection relaying that the *polar bear*, *horse* and *beaver* will continue to be fairly close (top of visualization) and the *gorilla* will be far away (bottom of visualization). We can also see three dimensions were increased to account for this modification: Jungle, New World and Old World. These dimensions explain the relative distances between the four highlighted points. All relative distances of the data points on the weight lines of the increased dimensions match the relative distances of the data points within the projection. The *polar bear* is similar to the *horse* and *beaver*, but different from the *gorilla*, when considering Jungle, New World and Old World. The user can visually see that the raw data values for the *polar bear*, *horse* and *beaver* are close together on the weight lines of the highly weighted dimensions. However, the raw data values for the *polar bear* and *gorilla* are far apart explaining the larger distance between them in the projection. Based on interview responses, all participants understood the relationship between relative distances between the visualization and the weight lines.

3.6 Stability

Problem. The two-dimensional layout of the high dimensional data is obtained through Weighted Multidimensional Scaling (WMDS) projection. Every time a user performs an object-level interaction, the layout of the points in the projection is updated. Each time a new layout is calculated, it can be vastly different from the layout the user created while modifying the locations of a subset of points because WMDS is rotation invariant. When this happens, it causes confusion and disrupts the user's analysis.

Solution. To maintain the persistence of the layout, we implemented an alignment function, which preserves the WMDS projected layout as close as possible to the user layout. This is achieved by ensuring that the positions of the user moved points stay as close to the same as possible between the user layout and the newly calculated layout. Preserving the original rotation decreases the effect of the rotation invariance of WMDS.

Results. Previous verbal feedback noted the abrupt change was confusing and disrupted the analysis. During the study, we did not receive such feedback, thus indicating the problem is fixed.

3.7 System Usability Scale Results

With this study, we find that our solutions to the usability problems worked. Participants answered the questions correctly and understood the new interactions. Based on the scoring of a SUS [1], Andromeda received an average rating of 81.78, which is interpreted as an A. This score means Andromeda is considered above average usability where the average is a score of 68. While using the tool, participants gave positive feedback noting it was easy and fun to learn, use, and interact with the interface.

4. DISCUSSION & CONCLUSION

This research has shed light on the existing gap between what a user is intending to perform with an interaction and how the model perceives this interaction. Through the usability study discussed in this paper, we find that interactions designed to bridge this gap can alleviate the issues cause by the gap.

To generalize these findings, the fundamental issue is about placing input and output *in context* of the metaphor in which the user seeks structure in the data. When considering WMDS in Andromeda, the context is the metaphor where proximity represents similarity. User input to interact with model parameters should be in context of the model output. For example, the highlight radius is shown to contextually select nearby points as relevant to the interaction. Likewise, output feedback should also be visualized in context. For example, showing the selected points on the weight lines helps reinforce the concept of weighted distance between the points. The effect is that the inputs and outputs of the model are closely connected in the user mental model. It should be contextualized in the user space instead of requiring the user to adjust separate widgets, which add burden to the user by requiring knowledge of the model parameters. Through observation of users performing analyses, we compiled a list of frequently occurring manipulations. This list includes the user intentions that need to match the model parameters.

- Dragging points to form a one or more new clusters (clusters sometimes based on outside knowledge, i.e., primates, rodents, vegetarians, etc.), to find relationships between some points;
- Dragging outlier into existing cluster, to find relationships between outlier and cluster;
- Maximize one dimension weight, to find extremes in reference to single dimension; or minimize one dim, to ignore it;
- Drag multiple sliders to equally large weights, then also highlight points to find what dimensions are related or correlated or to focus on groups of related dimensions based on outside knowledge.

By observing users working with the model, we learned their intents and designed appropriate interactions that bridge the gap between the user cognitive model and the mathematical model. Analytics by default is difficult, models are complex, and parameters are unintuitive. However, through good Human-Model Interaction research we can achieve good communication between humans and models, increasing model usability for human in the loop data analytics.

5. ACKNOWLEDGMENTS

This research was supported in part by NSF grants IIS-1447416, IIS-1218346, and DUE-1141096.

6. REFERENCES

[1] Brooke, J. SUS-A quick and dirty usability scale. *Usability evaluation in industry*. 189–194, 1996.

[2] Endert, A., Han, C., Maiti, D., House, L., Leman, S. and North, C. Observation-level interaction with statistical models for visual analytics. *IEEE Conference on Visual Analytics Science and Technology (VAST)*, 121–130, October 2011.

[3] Hartson, R. and Pyla, P. *The UX Book: Process and guidelines for ensuring a quality user experience*. Elsevier, 2012.

[4] House, L., Leman, S. and Han, C. Bayesian Visual Analytics: BaVA. *Statistical Analysis and Data Mining: The ASA Data Science Journal*. 8(1):1–13, 2015.

[5] Hu, X., Bradel, L., Maiti, D., House, L., North, C. and Leman, S. Semantics of Directly Manipulating Spatializations. *IEEE Transactions on Visualization and Computer Graphics*. 19(12):2052–2059, 2013.

[6] Kruskal, J.B. and Wish, M. Multidimensional Scaling. *Sage University Paper series on Quantitative Application in the Social Sciences*, 1978.

[7] Leman, S.C., House, L., Maiti, D., Endert, A. and North, C. Visual to Parametric Interaction (V2PI). *PLoS ONE*. 8(3), e50474, January 2013.

[8] Self, J.Z., House, L., Leman, S. and North, C. *Andromeda: Object-Level and Parametric Interaction for High-Dimensional Data Analysis*. Technical Report, Virginia Tech, 2016.

[9] Shneiderman, B. Direct Manipulation for Comprehensible, Predictable and Controllable User Interfaces. *Proceedings of the 2nd International Conference on Intelligent User Interfaces*. 33–39, 1997.