Designing Usable Interactive Visual Analytics Tools for Dimension Reduction

Jessica Zeitz Self
Department of
Computer Science
Virginia Tech
Blacksburg, VA 24061, USA
jzself@vt.edu

Xinran Hu
Bloomberg LP
New York City, NY 10022, USA
xhu102@bloomberg.net

Leanna House
Department of Statistics
Virginia Tech
Blacksburg, VA 24061, USA
lhouse@vt.edu

Scotland Leman
Department of Statistics
Virginia Tech
Blacksburg, VA 24061, USA
leman@vt.edu

Chris North
Department of
Computer Science
Virginia Tech
Blacksburg, VA 24061, USA
north@cs.vt.edu

Abstract
In this paper, we present principles focused on human-centered usability for developing interactive visual analytic systems that enable users to tweak model parameters directly or indirectly so that they may explore high-dimensional data. To exemplify our principles, we refer to our application, Andromeda, that implements interactive weighted multidimensional scaling (WMDS). Through its use, we uncovered design principles of effective, interactive, visual analytic tools. These design principles focus on two main areas: (1) the visualization and interaction and (2) the design of the communication between interface and algorithm.

Author Keywords
Interaction design; interface design; visual analytics; dimension reduction.

ACM Classification Keywords
H.5.2. Information interfaces and presentation (e.g., HCI): User Interfaces.

Introduction
All fields and domains require the use and analysis of data; however, not all domain experts are statisticians or algorithm experts. The nature of data analysis is challenging enough, but now researchers have to be trained in mathematical models that are supposed to simplify the analysis process. Statistical mathematical

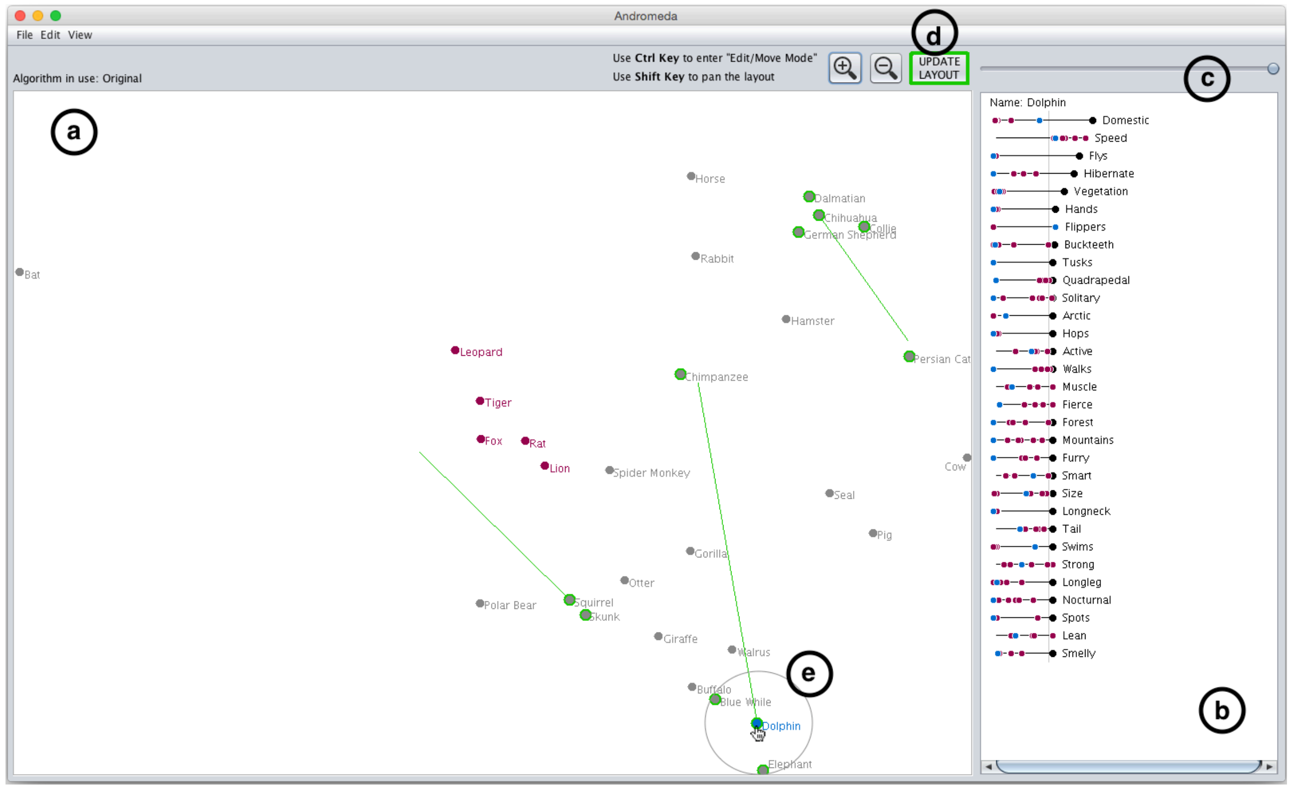

Figure 1: Andromeda interface exploring an animal dataset of 30 objects and 31 dimensions. Users manipulate points and dimensions to explore alternative projections: (a) the object view visualizes the high-dimensional data projected onto low-dimensional space using weighted multidimensional scaling (WMDS), and the user can drag points to explore alternative projections, (b) the parameter view displays all 31 dimension weights (as calculated by inverse WMDS) that the user can drag to reweight the importance of each dimension, (c) the slider tool animates the objects and weights transitioning from the old locations and values to the new locations and values, (d) the button updates the projection based on points the user has moved within the object view, and (e) the radius automatically highlights points in green for consideration by the algorithm as near and possibly important data points comparative to points being moved by the user. The current projection indicates that the cluster of red points (leopard, tiger, etc.) are less domestic, but faster than the blue point (dolphin) and its neighboring cluster of points.

models require in-depth expertise about parameters and other details to sufficiently use them. This adds a great burden to a researcher who solely wants to take advantage of these models to gain an understanding about and simplify her own collected data. What if we instead focus on providing an easier means to use the algorithms? Algorithms tend to be non-interactive, but if we make them interactive, we provide a means for users to relate to valuable algorithms despite their complex mathematical concepts [2].

Two types of interaction, parametric and observation level interaction (OLI), used within data visualizations that contain underlying models have been defined and shown helpful for data exploration [3,4,7]. Both forms of interaction enable users to adjust display-generating models directly and/or indirectly [1]. If the interaction methods are designed to be intuitive for a user and to fit the model being used, users have an increased chance of correctly and efficiently using the model on their data. However, there are constraints to these interactions and design considerations while developing

Figure 2: This is a sequence of interactions in Andromeda. (1) Initial view with moved points shown in green. (2)-(4) Updated layout with different clusters selected. This sequence indicates a three clusters: (1) non-vegetarian animals that range in size, (2) vegetarian animals that do not swim, (3) vegetarian animals that swim with flippers.

software with parametric and OLI capabilities that have not been formalized.

In this paper, we present principles to develop technically sound and useful visual analytic interfaces to exploit parametric interaction and OLI. We exemplify these principles in a tool we developed called Andromeda (see interface in Figure 1) [8]. It is a visual analytics tool that spatializes high dimensional data in two dimensions using an algorithm called Weighted Multidimensional Scaling (WMDS) [6]. We define 13 principles that designers should consider when developing these tools.

Design Principles

We established design principles that are necessary to consider when developing an interactive high-dimensional data visual analytics tool. These principles resulted from thorough evaluations of multiple iteratively-developed interfaces [8,9]. User challenges and misunderstandings gave rise to new design choices that better articulated the appropriate usage of the tool and the comprehension of the underlying model. We grouped the design principles into two categories: object and parameter visualization/interaction and algorithm communication.

Visualization & Interaction

The visualization provides a space to visualize and manipulate the data points as well as adjust any model parameters. In Andromeda, the parameters are the weights placed on all dimensions. We visualize each weight using a horizontal line with a handle at the end for adjusting the value of that weight. The raw data is displayed as points atop these lines strongly connecting the dimensions to the raw data for more efficient data exploration. The design principles in this section focus on the interactions and encodings necessary to utilize this space efficiently:

1. The projection visualization should utilize dynamic **object-level** and **parametric** interaction.
2. The interface should distinguish between view mode and edit/move mode.
3. Interactions should help align the user's mental model of input with the required algorithm input. The visualization should elicit and display all information possible from each user interaction to better inform the model. The visualization should automatically elicit more input from the user to the model and visually denote this input.
4. The interface should not require solely batch updates, but also automatic updates based on a singly moved object.
5. Any reasonable output from the model should be appropriately visualized in the interface as additional feedback to the user.
6. The interface should provide the user with incremental visual feedback in between model updates.
7. An interactive visual analytics tool should smoothly transition between updates.
8. Interactive visual analytics tools should retain object persistence between model visualization updates.
9. Interactions must adhere to the model constraints.
10. It is important to design an abstract way for the users to instinctively adjust the parameters without having to be experts about the model.

OLI [3] provides an intuitive interactive workspace for exploring complex high-dimensional data with the aid of powerful parameterized statistical models. Dimension

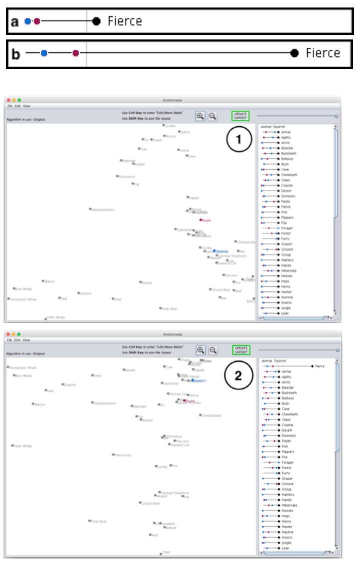

Figure 3: (a) and (b) show the difference in raw data point placement on the weight line after the Fierce dimension is increased. The two objects' relative distance in reference to Fierceness has increased since that dimension was emphasized. The Andromeda images depict: (1) the initial projection with all dimension weights equal, and (2) after a parametric interaction increasing the weight of Fierce. In each image, the squirrel is blue and the skunk is red in both the visualization and on the weight bars.

reduction algorithms (e.g. WMDS) reduce the data points to low dimensional space where users can naturally adjust these data points. **Object-level interaction** (OLI) allows the model to be hidden and instead relies on a familiar metaphor: **near is similar**. Users drag the points around the screen to form clusters, force outliers, or create other patterns. Points that get dragged closer are considered more similar and points that get dragged apart are deemed different. This feedback is pushed backwards through the algorithm to derive parameters, such as dimension weights, that would produce the desired projection. These derived parameters are then pushed forward again through the algorithm to show updates to the projection.

Along with this metaphor, **parametric interaction** allows users to adjust the underlying parameters that define the model. The type of model parameters should guide how they are displayed within the interface. Providing a visual representation of the numerical parameters may not make sense for all mathematical models, but designing an appropriate parametric interaction that decouples the interaction from the complexity of the parameters allows the user to focus on the data and not the model.

It is important to design tool interactions that are in keeping with the model constraints. Parameters must be contained within a feasible range of the parameter space. For example, dimension weights define the parameters of WMDS and are required to sum to 1. Because of this constraint, the parametric interaction of increasing a weight requires a decrease in other weights. As a visual cue of the model constraint, Andromeda dynamically decreases all other weights as a user increases a single weight.

The parameter view is scrollable to allow large numbers of visualized dimensions. However, to support fluid interactions and visualization updates, Andromeda sorts the dimension weights based on value from highest to lowest. This limits the amount of time the user has to spend scrolling through dimensions and also places the most important weights in the user's immediate view.

WMDS is also constrained by the real high-dimensional distances between the data points. However, these distances are altered when one dimension is emphasized over other dimensions, via dimension weights. Similarities and differences of the data points on a dimension are amplified when that dimension weight is increased. We overlay the raw data values on the weight lines to show the relative distances between data points as the weights are adjusted. For example, in Figure 3a, the squirrel and skunk have about the same level of fierceness. However, when fierceness is emphasized in Figure 3b (i.e. the fierce dimension weight is increased) either by parametric interaction or by OLI, the skunk appears relatively more fierce than the squirrel. Since the user increased the importance of the fierce dimension, the degree to which the animals differ will become more pronounced. This represents to the user how the algorithm interprets the data, and reinforces the metaphor that "near is similar."

It is important that the user understand the input she is providing to the algorithm as she manipulates points on the screen. Typically users move points with respect to other points. As the user is moving a point, nearby points are highlighted to reveal the importance of making these secondary points known to the algorithm (Figure 4). We designed the interaction to not only improve the results of the algorithm, but also to teach users how to specify the best input to receive insightful output.

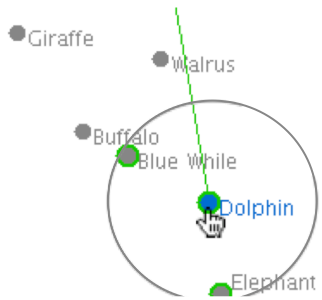

Figure 4: A user is moving the Dolphin data point. All points within the radius become highlighted, indicating that the user is moving the Dolphin with respect to those inside the radius. The green outline denotes a highlighted point which will be considered by inverse WMDS when deriving a new set of weights for this user modified projection. This helps the user learn that she must specify what other points she is moving the Dolphin with respect to. The radius provides a natural default that implicitly indicates that she is moving the Dolphin with respect to nearby data points. She can override these selections by clicking other data points instead. It is best not to select all points by default because that would likely over-specify the user's intent.

In Andromeda, by default the algorithm automatically updates after a single object has been moved. The user can turn off this feature and instead move multiple points before initiating the algorithm via a button. Dynamic feedback provides a fluid transition between layout updates and frees the user from deciding when the algorithm should recalculate.

The visual algorithmic feedback to the user is just as important as the input from the user. In order for the user to give feedback for the next iteration and continue the exploratory cycle efficiently, it helps to understand the algorithmic feedback. In Andromeda, with WMDS, algorithmic feedback includes: (1) a new layout, (2) the weighted dimensions applied to the real high-dimensional distance, and (3) a stress factor. The new layout and dimensions are displayed in the projection and parameter visualizations respectively. Stress can be displayed within the visualization to convey discrepancy between actual pairwise distances in the high-dimensional space and those plotted in the low-dimensional space [4,5,7].

Depending on the optimization calculation time, visual feedback might include a preview of the layout as the calculations are occurring. Another possibility might be to recalculate the layout using less strict constraints on the model just to give the user an idea of how the layout will change given the adjustments. The user could then decide whether to continue with the stricter model. All design choices should provide the user with feedback that represents all aspects of the mathematical model. This provides the opportunity for the user to better understand the data from many different angles.

Comparing the iterative visualizations is useful for analysis (see Figure 2). The visualization should smoothly transition between the updated results for continuity. In Andromeda, we designed a slider to rewind and replay the animations that automatically run between recalculations. This interaction allows users to manually trace the points' paths. However, because scale does not persist across iterations of WMDS, absolute distances cannot be compared between two iterations. This means that the distance between two points in one layout cannot be compared to the distance between the same two points in a future layout. Despite scale discontinuity between iterations, we decided this comparison was beneficial for the user to visualize the transition between states.

Algorithmic

Analytic tools with OLI capabilities have automated procedures in place to update display-generating parameters in response to specific user interactions. These procedures rely intimately on the models or algorithms chosen to generate the data visualizations. Some algorithms are more conducive for OLI than others. Principles of good algorithms include:

1. The algorithm should be invertible and incrementalized.
2. The inverted algorithm should emphasize explicit and consider implicit user input.
3. A non-deterministic algorithm should include additional implementation to mitigate stochastic effects.

Typical visual analytic algorithms rely on parameters to reduce data dimensionality for visualization purposes. Ideally, visualizations are functions of these parameters so that when visual adjustments are made, an inverted form of the function may solve for new parameter specifications keeping true to the model. For example, Andromeda relies on the algorithm, WMDS, whose parameters reflect the importance of each variable in a visualization in low-dimensional space. Our inverted optimization of WMDS must solve for these weights given a user-modified data point layout in the low-dimensional space. The result is a clear, quantitative, and parametric interpretation of changes in low dimensional coordinates in visualizations.

When calculating the weights, this inverted algorithm should not only emphasize explicit user input, but also consider implicit user input. When interacting with a screen full of objects, users tend to concentrate on a small number of objects. These explicit interactions contain more information about user semantics than the other objects on the screen [5]. Therefore, OLI systems should allocate more attention toward objects with which users have specifically interacted. This approach: (1) increases the likelihood of correctly identifying the semantics and (2) reduces the computational burden because it lessens the number of objects the model considers. Objects that are not directly manipulated by the user may still express user semantics. For example, a user may decide to move some objects toward a reference point (say Object A) in order to express similarity. Though Object A will be unmoved during the interaction, it is still of high value in understanding user semantics. Identifying these implicit objects is a tricky task. Two approaches should be considered: (1) provide appropriate interactions to assist users in being more explicit about their semantics and (2) nominate objects in close vicinity to an explicitly interacted object and allow the users to confirm or overrule these suggestions.

Crucial to OLI is that users may create multiple visualizations in a sequence that parallels their incremental sensemaking process. Thus, random perturbations in visualizations may confuse users; changes to visualizations should reflect added information provided by user interactions. If a user adds little information, the updated visualization should not drastically change because of peculiarities of the algorithm. Thus, stochastic algorithms or optimization schemes that may get stuck in varying locations due to function multi-modality may not be appropriate for OLI software, unless added precautions or steps are taken. For example, WMDS is invariant to scale, rotation, and reflection. To overcome this problem, Andromeda takes an extra processing step to align and scale coordinates with those from previous iterations. As a result new information in sequential visualizations is not masked by mathematical properties of WMDS.

Conclusion

We formulated design principles for visual analytics interfaces that encompass multiple views and ways of interacting with mathematical models for exploring high-dimensional data. Designers should consider all principles to fully understand the impact and interconnectivity of design choices within an interface. The goal of these principles is human-centered usability for machine-learning analytic algorithms.

We discussed the important role both object-level and parametric interaction plays in a well-designed visual analytics interface for exploring high-dimensional data. With both types of interaction, a user is able to gain more complex insights and accomplish new types of tasks [8]. In the future, we hope to see how these principles apply to other dimension reduction models so that they too are accessible to users without strong model knowledge.

Acknowledgements

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

References

1. Alex Endert, Lauren Bradel, and Chris North. 2013. Beyond control panels: direct manipulation for visual analytics. *IEEE computer graphics and applications* 33, 4: 6–13. http://doi.org/10.1109/MCG.2013.53
2. Alex Endert, Remco Chang, Chris North, and Michelle Zhou. 2015. Semantic Interaction: Coupling Cognition and Computation through Usable Interactive Analytics. *IEEE Computer Graphics and Applications* 35, 4: 94–99. http://doi.org/10.1109/MCG.2015.91
3. Alex Endert, Chao Han, Dipayan Maiti, Leanna House, Scotland Leman, and Chris North. 2011. Observation-level interaction with statistical models for visual analytics. *2011 IEEE Conference on Visual Analytics Science and Technology (VAST)*, IEEE, 121–130. http://doi.org/10.1109/VAST.2011.6102449
4. Leanna House, Scotland Leman, and Chao Han. 2015. Bayesian Visual Analytics: BaVA. *Statistical Analysis and Data Mining: The ASA Data Science Journal* 8, 1: 1–13. http://doi.org/10.1002/sam.11253
5. Xinran Hu, Lauren Bradel, Dipayan Maiti, Leanna House, Chris North, and Scotland Leman. 2013. Semantics of Directly Manipulating Spatializations. *IEEE Transactions on Visualization and Computer Graphics* 19, 12: 2052–2059. Retrieved October 24, 2013 from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6634115
6. J B Kruskal and M Wish. 1978. Multidimensional Scaling. *Sage University Paper series on Quantitative Application in the Social Sciences*.
7. Scotland C Leman, Leanna House, Dipayan Maiti, Alex Endert, and Chris North. 2013. Visual to Parametric Interaction (V2PI). *PLoS ONE* 8, 3: e50474. http://doi.org/10.1371/journal.pone.0050474
8. Jessica Zeitz Self, Leanna House, Scotland Leman, and Chris North. 2015. *Andromeda: Observation-Level and Parametric Interaction for Exploratory Data Analysis*. Blacksburg. Retrieved from http://infovis.cs.vt.edu/sites/default/files/jzself-tech-report.pdf
9. Jessica Zeitz Self, Nathan Self, Leanna House, Jane Robertson Evia, Scotland Leman, and Chris North. 2015. *Bringing Interactive Visual Analytics to the Classroom for Developing EDA Skills*. Blacksburg. Retrieved from http://infovis.cs.vt.edu/sites/default/files/cog_dim_tech_report.pdf