

Visual to Parametric Interaction (V2PI)

Scotland C. Leman¹✉, Leanna House^{1*}✉, Dipayan Maiti¹, Alex Endert², Chris North²

1 Department of Statistics, Virginia Tech, Blacksburg, Virginia, United States of America, **2** Department of Computer Science, Virginia Tech, Blacksburg, Virginia, United States of America

Abstract

Typical data visualizations result from linear pipelines that start by characterizing data using a model or algorithm to reduce the dimension and summarize structure, and end by displaying the data in a reduced dimensional form. Sensemaking may take place at the end of the pipeline when users have an opportunity to observe, digest, and internalize any information displayed. However, some visualizations mask meaningful data structures when model or algorithm constraints (e.g., parameter specifications) contradict information in the data. Yet, due to the linearity of the pipeline, users do not have a natural means to adjust the displays. In this paper, we present a framework for creating dynamic data displays that rely on both mechanistic data summaries and expert judgement. The key is that we develop both the theory and methods of a new human-data interaction to which we refer as “Visual to Parametric Interaction” (V2PI). With V2PI, the pipeline becomes bi-directional in that users are embedded in the pipeline; users learn from visualizations and the visualizations adjust to expert judgement. We demonstrate the utility of V2PI and a bi-directional pipeline with two examples.

Citation: Leman SC, House L, Maiti D, Endert A, North C (2013) Visual to Parametric Interaction (V2PI). PLoS ONE 8(3): e50474. doi:10.1371/journal.pone.0050474

Editor: Fabio Rapallo, University of East Piedmont, Italy

Received: April 4, 2011; **Accepted:** October 24, 2012; **Published:** March 20, 2013

This is an open-access article, free of all copyright, and may be freely reproduced, distributed, transmitted, modified, built upon, or otherwise used by anyone for any lawful purpose. The work is made available under the Creative Commons CC0 public domain dedication.

Funding: This work was supported by National Science Foundation, Computer and Communications Foundations, #0937071. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

* E-mail: lhouse@vt.edu

✉ These authors contributed equally to this work.

Introduction

Organizing and understanding large datasets are complex tasks for many scientists, engineers, and intelligence analysts. To aid them in such sensemaking endeavors, tools have been developed to visualize high-dimensional data. These tools rely on mathematical models or algorithms that collapse high-dimensional data matrices to much smaller visual spaces (i.e., spaces of only two or three dimensions). For example, common visualizations of high-dimensional text data extend upon a geography metaphor and use algorithms to display such data in two-dimensional maps [1]. One problem is that visualizations can mislead users, just as any data summary might, by over-simplifying features or structures in high-dimensional datasets. Therefore, low-dimensional versions of high-dimensional data have the potential to be misleading. When this happens, users currently have limited options to correct the problem.

Namely, displays of data in two or three dimensions result typically from a linear visualization pipeline shown in Figure 1, where data D are summarized by a mathematical model or algorithm $M(\theta)$ first and subsequently mapped to a visual display V . A display is controlled solely by the algorithm that generated it and adheres to predefined mathematical objectives, constraints, or parameters denoted by θ . When these constraints contradict expert judgment, they warp or miss useful data features and visualizations can lose interpretability. For examples, consider visualization methods Principal Component Analysis (PCA) [2] and Multidimensional Scaling (MDS) [3]. PCA is a common analytical approach that projects datasets to two dimensions (in the case of visualization) in the directions with the highest variance. PCA loses its utility when meaningful features in the data do not

correspond with variance. Similarly, MDS is an analytical approach that solves for low-dimensional (e.g., two-dimensional) coordinates of data points by minimizing the difference between pairwise distances of observations in high- and low-dimensional spaces. When the chosen distance function lacks relevance to the application, MDS can produce visualizations that are hard to interpret.

As defined by the current pipeline (Figure 1), users do not have an intuitive or natural means to correct visual inaccuracies-beyond the option of starting the pipeline over. For example, users can transform the data or adjust the display-generating model (e.g., tweak model parameters θ) to re-implement the pipeline and create new visualizations. This means that users, who may not have the appropriate mathematical training, must have a deep enough understanding of the display-generating models to change them or the data in a way that will result in useful visualizations. When users cannot parameterize their expert judgements, the pipeline is broken and sensemaking stalls.

In the field of Visual Analytics (VA), the disconnect between static displays of data and usability has been studied extensively and has motivated research in human-computer interaction [4–7]. It has been shown that when users interact with visualizations, users learn more from the data than when they do not, even when the model is arguably poor. In the Methods Section, we define common forms of interaction that are readily available in many VA tools: surface-level and parametric interaction. VA tools that enable surface-level interactions allow users to edit displays independent of the underlying models or algorithms; e.g., highlighting or filtering observations. Whereas, when users interact parametrically, they manually change influential parameters in the display-generating models or algorithms. For example, iPCA [8]

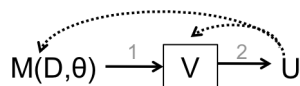


Figure 1. Standard visualization pipeline. Data D feeds into a mathematical model M that relies also on parameters θ , and produces a visualization V . The users U make sense of the visualization to the best of their abilities. To correct any visual inaccuracies, users must either change M , D , or θ .

doi:10.1371/journal.pone.0050474.g001

and XGvis [9] are VA tools that allow users to adjust dials or sliders that either augment or out-right change influential parameters in PCA or MDS, respectively.

Despite the extensive development of human-computer interactions and VA tools, sensemaking of complex data is still limited. The pitfall of current forms of interactions is that users are still constrained by the linear pipeline and placed at either the beginning or end. For example, surface-level interactions take place at the end of the pipeline by users who are trying to salvage information from a potentially misleading view of the data. Whereas, parametric interactions take place at the beginning of the pipeline by users who are forced to make model adjustments. Although parameter-controlling dials and sliders are easy to manipulate, users must still understand the mathematical models or algorithms to explore data efficiently. Without understanding the mathematics, users have two options a) hope (not know) that their parametric adjustments will convey their expert judgments appropriately or b) evaluate data visualizations given every possible combination of the parametric settings. For one parameter (e.g., one dial), option b) is do-able, even when the parameter is continuous. However, given two or more continuous parameters, we think that option b) will quickly overwhelm users by the infinite number of parameter combinations.

In this paper, we discuss a new form of human-computer interaction to which we refer as “Visual to Parametric Interaction” (V2PI). We recognized that when users make certain surface-level changes to displays, the users are communicating that the display-generating algorithm is not working properly. V2PI interprets quantitatively what is communicated by the users to make parametric model changes (and, subsequently, new visualizations). In Endert et al. [10], we provide specific examples of V2PI. Here, we not only apply V2PI, but also define V2PI explicitly, highlight the framework to develop a V2PI, discuss advantages and disadvantages of V2PI, and explain fundamental changes in the process to visualize and explore data when V2PI is possible. V2PI transforms data visualizations from being static to dynamic in that V2PI enables information to flow fluidly between display-generating algorithms and users. Users learn from visualizations and the visualizations adjust to expert judgement. Thus, with V2PI, the visualization pipeline becomes bi-directional in that users are not simply at the starting nor receiving end of the pipeline, but are embedded in the visualization scheme formally.

To be clear, VA tools that enable V2PI and rely on the bi-directional visualization pipeline foster data exploration. V2PI does not guarantee the discovery of all or any particular feature in the data. As a data exploration tool, V2PI merges two learning technologies: 1) statistical/data mining methods and 2) interactive visualization techniques. The first technology focuses on mathematical/algorithmic representations of data, whereas the second provides cognitive representations of data. While V2PI maintains the rigor of mathematical/algorithmic technologies, users only operate within visual layouts of data. Hence, again, the methodology we develop from the merger is one for data

exploration. In our examples (Results Section) we reiterate the exploratory nature of our methods, but for explanatory reasons we offer a “ground-truth” to exemplify how visualizations adjust and what we can learn using V2PI.

The remainder of the paper has four main sections: Methods, Results, Discussion, and Conclusion. In the Methods Section, we provide background about visual analytic interactions and introduce V2PI. We define V2PI, develop the bi-directional visualization pipeline, and explain required steps to construct V2PI VA tools. In the Results Section, we apply V2PI in two case studies. For each case study, we describe the data at hand, a reasonable method for visualizing it, potential feature-masking constraints of the methods, and implement V2PI to relax those constraints. We reflect on the case studies in the Discussion Section to acknowledge both the benefits and limitations of V2PI. In the Conclusion Section, we summarize our current and future work.

Methods

2.1 Background: Visual Analytic Interactions

The process of using data to update domain specific knowledge is referred to as sensemaking [11,12] and has been represented in the form of a sensemaking process [13,14]. In this process, analysts (i.e., experts, users, applied researchers, etc.) begin with a knowledge base that they hope to either expand or adjust given the data. The information discernible in data is often unclear to analysts. Thus, learning from data may take place over time or a series of iterations during which analysts explore the data and assimilate what they observe with their knowledge bases. Such explorations/assimilations may take place each time analysts interact with data.

In fact, Pike et al. [7] states, “interaction is the insight,” and according to Thomas and Cook [12], Visual Analytics (VA) “is the science of analytical reasoning facilitated by interactive visual interfaces.” In VA, various types of interactions have been studied, and Pike et al. [7] categorize them into two main groups: *lower-* and *higher-* level interactions. The primary difference between these groups pertains to the goal of the users when they interact with the data. With lower-level interactions, users aim to summarize “low-level structure” in the data including maxima, minima, simple patterns, and linear trends. Examples of such interactions include filtering, sorting, and other specific formal queries. Any interactions that are not considered lower-level are higher-level. The purpose of higher-level interactions is to “understand” the data by uncovering features based on abstract or complex (e.g., nonlinear) data characterizations.

In this section, we refine the interaction groups further as *surface-level* and *parametric* to motivate the development of V2PI. We explain each of the interactions within the context of Figure 2. Figure 2 was created by a VA tool called IN-SPIRE [15] and displays a “Galaxy View” of text data that were collected for an intelligence analysis. In this spatialization, the data points, i.e., documents, are represented by dots and clustered algorithmically by IN-SPIRE. The aim for IN-SPIRE is to assist users in grouping similar documents together and displaying them in an accessible fashion.

2.1.1 Surface-Level Interactions. Surface-level interactions are performed purely within the visual domain and are contained in the lower-level class of interactions. Data rotations, reflections, and translations, highlighting or editing observations, and zooming into a portion of the visual space are each examples of surface-level interactions. These interactions, while capable of enhancing the understanding of complex data structures, do not necessarily relate coherently to mathematical data structures.

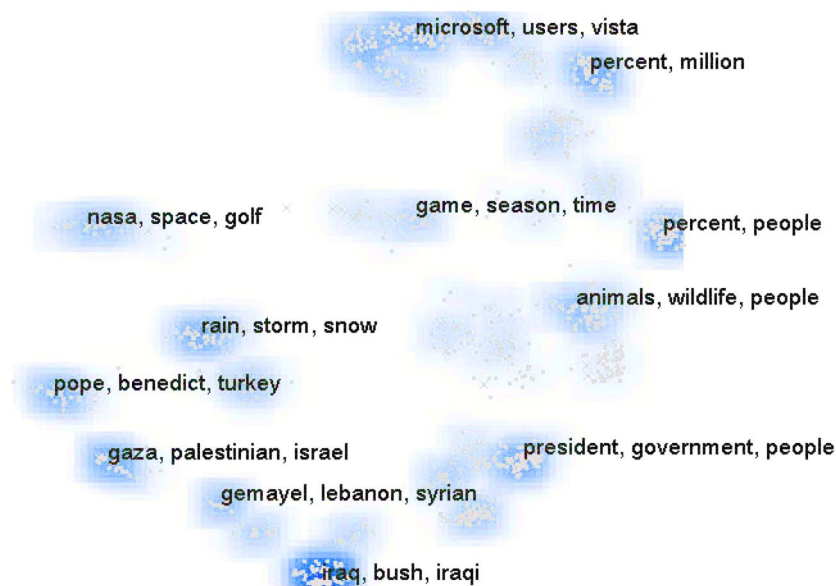


Figure 2. A “galaxy view” of text data created by the IN-SPIRE suite of data visualizations. In-SPIRE uses complex mathematical models in order to discern structure (e.g., clusters) in high-dimensional data.
doi:10.1371/journal.pone.0050474.g002

Within the context of Figure 2, surface-level interactions may include opening, closing, highlighting, and filtering documents or repositioning clusters. For example, users may wish to drag the cluster labeled by, “rain, snow, storm,” to the bottom right of the screen because they feel that the cluster is unimportant. This adjustment is independent of the underlying algorithm and committed purely for organizational purposes.

2.1.2 Parametric Interactions. Parametric interactions are performed directly on the mathematical models that control visualizations. iPCA and XGvis are VA tools that permit parametric interactions; iPCA allows users to interact directly with the principle eigenspace of the data, and XGvis enables users to change either the analytical metric scaling method (measure for distance between observations) or the local optimization scheme used to solve for lower dimensional versions of high-dimensional observations. If IN-SPIRE had the capability for a user to specify, say, the number of clusters in Figure 2, it would be an example of a tool that also permits parametric interactions. Table 1 provides a non-exhaustive list of other parametric interactions.

Regardless of whether users apply parametric or surface-level interaction, they are often trying to match the visualization to their personal mental maps of the data. A user is more likely to make sense of the data when the data appear in an expected form. However, editorial changes to visualizations dismiss their mathematically driven interpretations, and, parametric changes may not produce ideal visualizations for users. Mental maps of data

may not comply to rigid, parametric characterizations of the data. This means that regardless of how many times parameters are adjusted, it is possible that suitable images of data may never be obtained by users. What users need is an interaction that balances surface-level and parametric adjustments to displays of data. For this reason, we develop Visual to Parametric Interactions (V2PI)

2.2 Visual to Parametric Interactions (V2PI)

Surface-level interactions are intuitive to implement, but may lack analytical interpretation because they are independent of the mathematical underpinnings of visualizations. Parametric interactions maintain the integrity of mathematical data characterizations, but can be difficult for analysts to understand. To combine the ease of surface-level interactions and the mathematical rigor of parametric interactions, we introduce Visual to Parametric Interaction (V2PI). In this section, we define Visual to Parametric Interaction (V2PI), show how it fits in a bi-directional visualization pipeline, and refine technical points about V2PI. Subsequently, in the Results and Discussion Sections respectively, we apply and summarize V2PI in case studies.

2.2.1 Definition. V2PI is the act of making surface-level interactions that are interpreted by software quantitatively to make parametric model changes (and, subsequently, new visualizations). For example, one interpretation of the clustering structure in Figure 2 is that observations within clusters are more correlated than observations between clusters. Suppose a user chose to

Table 1. A non-exhaustive list of parametric interactions.

Visualization	Parametric Interactions
Data in clusters	A user defines a cluster by specifying the required shape, minimum distance from other clusters, or minimum number of elements.
Data network	A user adjusts the number of nodes and/or edges.
Classification tree diagram	A user adjusts the probabilities that branches split.

doi:10.1371/journal.pone.0050474.t001

commit a surface-level interaction by merging two neighboring clusters together. This interaction suggests that the algorithm (as parameterized) underlying the IN-SPIRE visualization underestimates the correlation between a subset (those selected) or all observations. If IN-SPIRE had V2PI capabilities, IN-SPIRE would quantify and parameterize the merger to adjust all or a subset of pairwise correlation measurements. In turn, IN-SPIRE would use the adjusted correlation measurements to create a new display with different clusters and ready for further V2PI.

The novelty of V2PI is that developers of VA tools with V2PI functionality must learn users' intent from surface-level interactions and develop a strategy to automate mathematical adjustments to display-generating models accordingly. Thus, developers must know, in advance, how to interpret, process, and parameterize various surface-level interactions. Table 2 provides a non-comprehensive list of surface-level interactions with possible parametric interpretations. Alas, not every surface-level interaction will have a meaningful parametric interpretation and, for those that do, the process to parameterize the surface-level interaction is model specific. We discuss the process in the Section 2.2.3, Parameterizing Feedback, and provide examples in the Results Section.

The primary advantage of V2PI is that displays of data that were once static become dynamic. They can respond indefinitely to surface-level interactions (with the parametric interpretations) to account for expert judgment and potentially reveal additional information in the data. Thus, as users learn more, new visualizations can update accordingly; and, as visualizations update, users can learn more. With V2PI, is a bi-directional flow of information in the visual domain of the data between display-generating models and users. In the next section, we develop the concept of a bi-directional visualization pipeline in detail and further explain V2PI.

2.2.2 V2PI and the Bi-directional Pipeline. By construction, visualizations that result after a user's V2PI are dynamic and represent both the high-dimensional data according to the model or algorithm and expert judgement. Users learn from the visualizations and the visualizations adjust to user feedback, as defined by the parametric interpretation of some surface-level interactions. By interpreting the interactions in a parametric form, a) the models or algorithms work as defined originally, but now

rely on both the data and user feedback; and b) the models create new visualizations that are subsequently available for additional feedback.

To see this, consider Figure 3, a bi-directional version of the original visualization pipeline. This version is similar to Figure 1, except users may now receive and distribute information in the visualization iteratively. Specifically, **Steps 1** and **2** of the bi-directional pipeline are similar to the original in that a mathematical model M that relies on data D and parameters θ constructs a visualization V that users U assess for sensemaking. Now, with V2PI, users have the opportunity to commit either standard (surface-level or parametric) interactions or offer feedback about the model via the visualization. If users choose V2PI, they make surface-level adjustments to visualization V to create V' (the original V with adjustments). We distinguish standard surface-level interactions from those associated with V2PI by referring to the latter as *cognitive feedback*, F_c . That is, in **Step 3** of the bi-directional pipeline, users communicate their F_c by creating visualization V' . In **Step 4**, the cognitive feedback is parameterized to update θ for $M(D, \theta)$ accordingly; we refer to the parameterized version of F_c as *parametric feedback* F_p . This step is represented by a dashed line because, in practice, users are protected from the parameterization of F_c . VA developers of visualization tools with V2PI capabilities must have the computational and mathematical machinery in place to parameterize cognitive feedback. Given F_p and the updated θ , the pipeline steps may repeat.

Steps 1–4 may iterate until any of the following occurs: users are satisfied with the display; the data have been explored thoroughly; or the sensemaking process is complete. For this reason, the bi-directional pipeline is similar in spirit to typical depictions of sensemaking and human-computer interaction, including those developed by Norman, Abowd and Beale, and Keim et al. [5,16,17]. Such depictions outline actions that need to be taken by the user and/or the system (e.g., data analysis, computer, visualization, etc.) to enable sensemaking of data. The bi-directional pipeline, however, is more detailed than these depictions. For example, Keim et al. [5] present an iterative "Visual Analytics Process" that considers the potential for users to obtain insight from visualizations and commit interactions "to

Table 2. A non-exhaustive list of V2PI.

Visualization	Surface-Level Interaction	A Parametric Interpretation
Data in clusters	Move two points from different clusters to the same cluster	Up weight the current clustering role of the dimensions in which the observations are similar
Two-dimensional map or spatialization of data	Change the relative locations of points	Down weight the dimensions that dictate the current map
Data network across nodes/data points	Delete a connection between nodes	Decrease the current correlation between the nodes
Classification tree diagram	Delete a classification branch	Reduce the current marginal probability of belong to the corresponding class

V2PI requires parametric interpretations of surface-level interactions.
doi:10.1371/journal.pone.0050474.t002

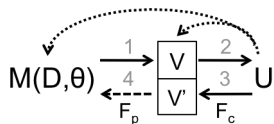


Figure 3. The bi-directional visualization pipeline. Step 1) Create visualization V based on a mathematical model or algorithm M that depends on data D parameters θ ; Step 2) display the visualization for users U to assess, Step 4) Users adjust the visualization to offer model feedback; and Step 5) Update the model M (e.g., via the parameters θ). doi:10.1371/journal.pone.0050474.g003

refine parameters in the analysis process and steer visualization.” Whereas, the bi-directional pipeline describes specifically how users interact with displays of data and how the system interprets these interactions to update the analytical process, when the parametrization machinery (Step 4) is in place. Also, communication between the system and the users in the bi-directional pipeline takes place explicitly in the data visual domain. In fact, visualizations V and V' are connected in Figure 3 to emphasize this point.

The process to parameterize cognitive feedback is model and application specific. Unlike standard constraint-based user interfaces that are described in Myers et al. [18], new visualizations in the bi-directional pipeline do not simply result from fixing adjustments in V' and configuring what remains in the visualization accordingly. Rather, we learn from V' how we might adjust display-generating parameters that would impact the entire display jointly. Careful thought is needed to interpret and quantify cognitive feedback in a form that *both* captures the users’ intent (reasons for injecting the cognitive feedback) and is compatible with the model. In the next section, we highlight what needs to be considered when parameterizing feedback and, in subsequent sections, we provide examples within the context of case studies.

2.2.3 Parameterizing Feedback. In both the original and bi-directional visualizations pipelines, the visualizations depend upon a model M with inputs, data D and parameters θ . If we consider the data D to be given (e.g., we do not transform nor filter the data), visualizations can only change when we alter specifications for θ . Thus, in some sense, all visualizations rely on potentially tunable parameters θ . Within the context of V2PI, we parameterize feedback that is communicated by V' to tune specifications for θ ; i.e., we use F_p to adjust the model parameters from an original setting, $\theta = T$, to a setting that accounts for feedback, $\theta = T^F$. In turn, new visualizations rely on the model M , data D , and expert-adjusted parameter specifications $\theta = T^F$. For example, some models M rely on an optimization procedure to set $\theta = T$ and visualize data D . Based on F_p , we might adjust the procedure which will subsequently result in calculating $\theta = T^F$.

The challenge is formulating F_p from V' so that we can specify T^F . Our solution is two fold. First, we solve an inverse problem in that we estimate a value for θ that would result in either the adjusted display V' or the adjusted observations within V' . This solution is F_p , a parametric interpretation of F_c . Second, we take a weighted average of T and F_p to set T^F ,

$$T^F = \rho F_p + (1 - \rho)T, \quad (1)$$

where $\rho \in [0,1]$ and ρ reflects the weight users want to place on their judgements relative to the current visualization, e.g., when $\rho = 1$, θ is specified entirely by expert judgement in that $T^F = F_p$.

The choice to take a weighted average of T and T^F is both flexible and justifiable theoretically when θ is assessed using Bayesian methods [19]. If users are unclear about weight ρ , they may apply parametric interaction to observe how their feedback impacts a visualization by slowly transitioning ρ between 0 and 1.

We have mentioned several times that V2PI may occur in sequence; i.e., the bi-directional pipeline may repeat several iterations before a user feels satisfied with the data exploration. With each injection of cognitive feedback, a parametric form is derived and a new visualization is created. To convey this mathematically, consider the i th execution of V2PI such that

$$T^{F,i} = \rho^i F_p^i + (1 - \rho^i)T^{F,i-1}, \quad (2)$$

where $T^{F,i-1}$ represents the specification for θ that created the visualization which was adjusted for the i th iteration and $T^{F,0} = T$ (the original specification for θ). There is no notion of convergence when considering V2PI. Users choose to stop iterating when the data visualizations make sense. In some cases this means that users may stop when a particular structure in the data appears or, in other cases, when users assess the data from multiple perspectives (based on multiple implementations of V2PI) and simply feel comfortable with the data exploration. For the sake of being clear about V2PI, we exemplify it and the bi-directional pipeline in the next section using case studies that fall into the former category.

Results

We provide two case studies that rely on either PCA or MDS to demonstrate the development and use of V2PI. PCA and MDS are similar (in fact, under some conditions, the same) in that each produces a spatialization of data for which the relative pairwise distances between observations has meaning; observations that appear close or far apart in visualizations are similar and different, respectively, in the high-dimensional data spaces. Thus, each case study allows users to explore data and adjust the coordinates of two or more observations (hence change the relative distances between points) to communicate cognitive feedback in the bi-directional visualization pipeline. However, PCA and MDS can differ by the way they learn the low-dimensional relative distances between observations. In turn, the methods we use to interpret and parameterize the cognitive feedback in the visualizations are different.

We begin each case study with a description of data and theoretical details of the analytical procedures, PCA or MDS. Then, we use the steps of the bi-directional pipeline to guide our discussions. We develop V2PI based on the parameterization of one form of cognitive feedback per example.

3.3 Case Study 1: PCA

3.3.1 Data. The bi-directional pipeline and V2PI fosters data exploration and has the potential to reveal structure (when it exists) in data spatializations, such as clusters. To begin an exploration, experts often use what they know about the data. However, what they “know,” may be incomplete or reflect mere conjectures. For example, in genetic analyses, biologists might know the pathways to which some genes belong, but not all; or, to assess voting tendencies, political analysts might know the party affiliations of some voters, but not all. For such cases, it is reasonable to take semi-supervised analytical approaches to assess data and infer the global data structures. In this section, we use simulated data to emulate such scenarios.

We simulated a $p=3$ dimensional data set \mathbf{x} that contains $n=300$ observations and three clusters, as shown in Figure 4A. Since we simulated the data, we have access to detailed information concerning the cluster assignments of each observation. However, we only reveal the cluster assignments for 20 of the 300 observations; ten observations were selected at random from clusters 1 and 2 each. To visualize these data we apply PCA and highlight the selected observations in Figure 4B. Notice in Figure 4B that observations from clusters 1 and 2 do not group. Rather, they are mixed in a seemingly random scatter within the remaining data. Based solely on the display, we cannot make

judgements about, say, the number of clusters, size of clusters, and assignments of observations to clusters.

If we were willing to use the true classifications for the remaining 290 observations, we could define the clusters as a function of the dimensions in \mathbf{x} by using fully supervised learning strategies, such as a labeled version of PCA or linear discriminant analysis [20,21]. However, we consider only what is known about the 20 highlighted observations in Figure 4B and we take a visual data exploration approach. In the sections that follow, we develop both the mathematical and computational machinery to apply

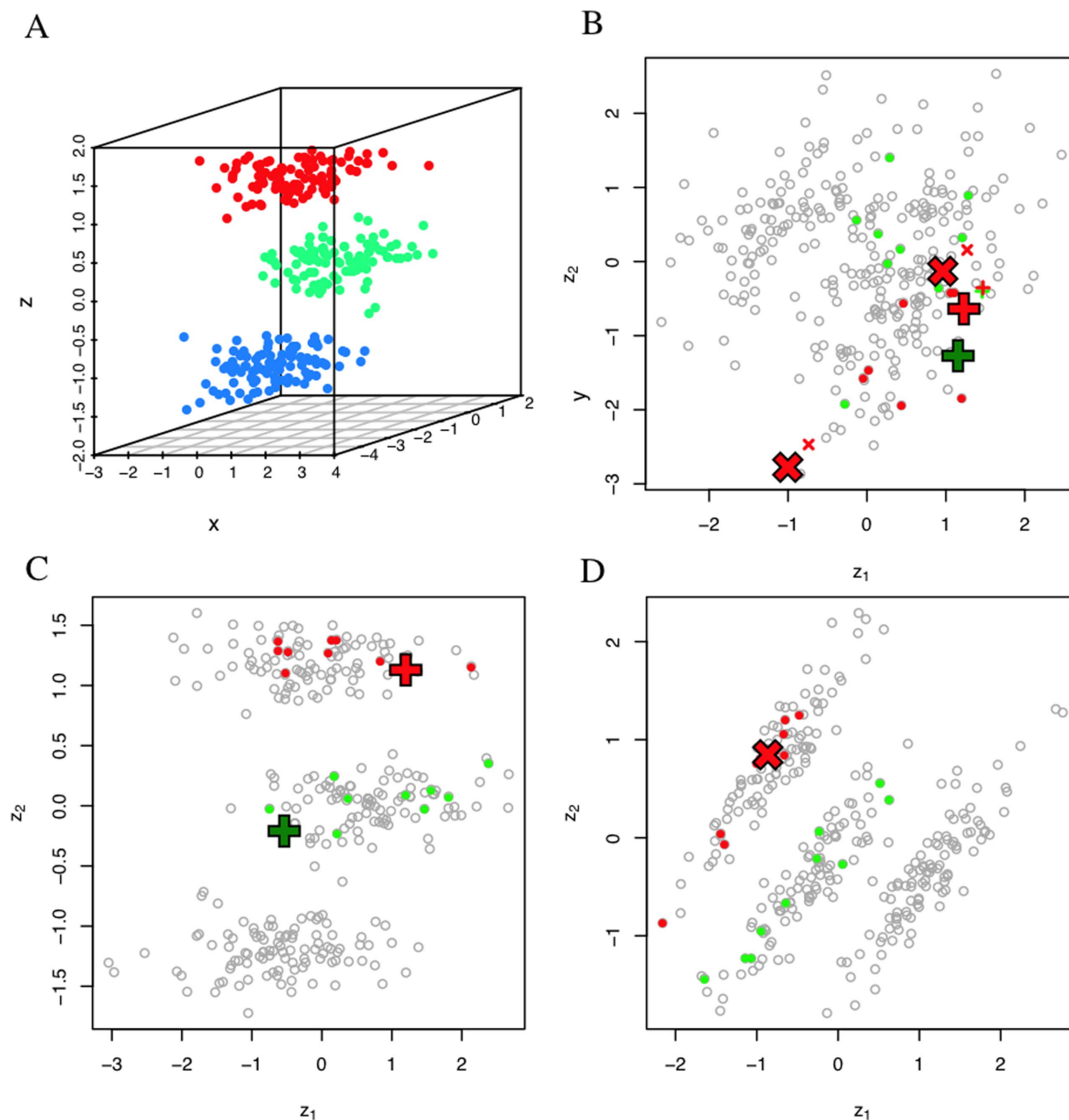


Figure 4. V2PI with PCA. Figure A displays the simulated data in three dimensions. Observations in red, green, and blue denote groups 1, 2, and 3 respectively. Figure B displays the PCA projection of the simulated data with 20 observations (that were selected at random) highlighted. Again, red and green points represent observations in groups 1 and 2 respectively. Figures C and D show updated displays after an adjustment to Figure B. Figure C is the result of moving points marked by '+' in Figure B apart and Figure D is the result of moving the points marked by 'x' in Figure B together. Notice that both adjusted visualizations capture the clustering structure. doi:10.1371/journal.pone.0050474.g004

V2PI and create new PCA visualizations. We start first by explaining the technicalities of PCA.

3.3.2 Description of PCA. PCA is a deterministic, analytical procedure that relies on an optimal linear projector to reduce the dimension of a data set. Consider a center-shifted, p -dimensional data set \mathbf{x} that contains observations \mathbf{x}_i where $i \in \{1, \dots, n\}$; i.e., $\mathbf{x} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ and \mathbf{x} is $p \times n$. In our simulated example above, $p=3$ and $n=300$. PCA relies on the solution for a $q \times p$ transformation matrix \mathbf{W} , where $q < p$, that maximizes the variance of a low-dimensional version of \mathbf{x} which we denote by \mathbf{z} .

To solve for \mathbf{W} , one option is to take the eigen-decomposition of the sample variance (of \mathbf{x}), \mathbf{S} , such that $\mathbf{S} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T$, where \mathbf{U} is $p \times p$ and contains the eigenvectors of \mathbf{S} , $\mathbf{U} = \mathbf{V}$, and $\mathbf{\Lambda}$ is a diagonal matrix that includes the ordered eigenvalues of \mathbf{S} (e.g., the element in the first column and row of $\mathbf{\Lambda}$ contains the largest eigenvalue of \mathbf{S}). Since the eigenvectors that correspond to the two largest eigenvalues determine the two orthogonal directions that explain the most amount of variance in \mathbf{x} , \mathbf{W} is assigned to equal the first two columns of \mathbf{U} ,

$$\mathbf{W} = \mathbf{U} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0_{p-2} & 0_{p-2} \end{bmatrix},$$

where 0_{p-2} represents a $(p-2) \times 1$ vector of zeros. Given \mathbf{W} , the calculation for \mathbf{z} is straightforward,

$$\mathbf{z} = \mathbf{W}'\mathbf{x}. \quad (3)$$

When $q=2$, a PCA visualization simply plots \mathbf{z} (e.g., Figure 4B) in a two-dimensional scatterplot. The axes of the plot are hard to interpret, but, fortunately, it is only the configuration of the points in the plot that matters. PCA spatializes observations so that the relative distance between them reflects their relative similarity in the dimensions most preserved. As defined by the current form of PCA, these dimensions are those with the largest variances. Alas, because of PCA's strict variance criteria and explicit assignment of \mathbf{W} , the spatialization can mask structures in data that do not correspond with variance. For example, the within-cluster variance is larger than the between-cluster variance in the data shown in Figure 4A. Thus, the clusters do not appear in \mathbf{z} as plotted in Figure 4B, and despite knowing the presence and/or characteristics of the clusters, we cannot adjust \mathbf{W} .

In the next section, we transform PCA from a deterministic, dimension reduction algorithm to an expert guided projection method via V2PI and the bi-directional pipeline. We explain within the context of the data set described in Section 3.3.1. The goal is to allow experts to explore data from different perspectives using PCA methods so that the clusters may (or may not) be revealed. The advantage is that the interpretation of each data spatializations from the different perspectives is maintained (i.e., relative distances between observations reflect relative similarity), but structures that do not depend on variance have the potential to be discovered.

3.3.3 PCA with V2PI. We start by applying PCA for Steps 1 and 2 in Figure 3. We derive \mathbf{W} in accordance with Equation (3) and display \mathbf{z} as we did in Figure 4B. For Step 3, experts participate in the data analysis by assessing and injecting feedback F_c about the projection. Since the configuration of points has meaning in data spatializations, a natural surface-level interaction F_c to parameterize is a reconfiguration of the points. Here, we

develop V2PI so that users may re-configure the location of two observations; i.e., to create \mathbf{V}' , users may either drag two observations together or apart. The choice to drag observations together or apart depends upon expert judgment. If an expert believes two observations are similar in the high-dimensional space, but they appear distant in the visualization, the expert may drag the observations together. Whereas, if an expert believes two observations are different in the high-dimensional space, but they appear close together in the visualization, the expert may drag the observations apart. For example, in Figure 4B, a user may choose to drag two observations from cluster 1 (in red) together, two observations from cluster 2 (in green) together, or one observation from each cluster apart.

As a VA tool developer, we could have developed a more complex version of V2PI; e.g., allow users to move many observations. However, what we propose is still a viable form of V2PI and helps to convey the relative simplicity for how to use V2PI. Namely, experts need only have knowledge about the relationship between two observations to re-assess data from a different perspective. They do not need to have reliable judgments concerning, say, the dimensions in the data that define clusters; the number of clusters in the data set; nor the size of data clusters. Also, the methods can be extended to allow cognitive feedback with more than two observations. In fact, for the next case study, we do just that based on MDS (an analytical method that can reproduce PCA plots under some constraints); we allow users to move several observations to communicate cognitive feedback.

To parameterize F_c **Step 4** of the PCA bi-directional pipeline, we must a) determine a user's intent and b) represent it in a quantitative form that is compatible with PCA. When users drag observations together, the users are suggesting the need for a display that up-weights the dimensions for which the observations are similar and down-weights the dimensions for which they are different; whereas, when users drag observations apart, the users are suggesting the need for a display that up-weights the dimensions for which the observations are different and down-weights the dimensions for which they are same. For PCA, the dimensions that have relatively large and small weights are those with a high and low variances respectively-the transformation matrix \mathbf{W} results from deterministic procedure based on the sample variance \mathbf{S} . Thus, depending upon F_c , we re-weight the elements of variance matrix \mathbf{S} accordingly.

To do so, we derive a distance matrix as F^p that is both indicative of the observation adjustments and similar in nature to a data variance matrix in that it is $p \times p$ and semi-definite. We describe one procedure for deriving the distance matrix from F_c in File S1. Given F_p , we take a weighted average as described in Section 2.2.3 to calculate \mathbf{S}^F ,

$$\mathbf{S}^F = \rho F_p + (1 - \rho)\mathbf{S}.$$

For a new visualization, we re-apply the PCA machinery; i.e., we determine the transformation matrix \mathbf{W} based on \mathbf{S}^F and recalculate \mathbf{z} as defined by Equation (3).

We provide two adjusted PCA visualizations in Figure 4. Figures 4C and 4D are based on the cognitive feedback that two observations were dragged together and apart in step 3, respectively. Notice that regardless of the action taken for F_c , the adjusted figures display structure. In fact, from injecting information about the relationship between two observations, we learn from the updated view of the data that 1) the data include three clusters and 2) the cluster-assignments of every observation.

3.4 Case Study 2: MDS

3.4.1 Data. In the previous case study, we used a simulated example to show how V2PI works. Now, we consider a more realistic dataset \mathbf{x} that describes $n=25$ cities (Amherst, Ann Arbor, Atlanta, Atlantic City, Blacksburg, Bloomington, Boston, Chapel Hill, Charlotte, Chicago, Davis, Denver, Detroit, Fort Collins, Helena, Houston, Knoxville, Los Angeles, Miami, New York City, Reno, San Francisco, Seattle, Tucson, and Washington D.C.) based on ten variables: Latitude, Longitude, Income (median), Age (median), Population, Housing price (median), Population density, Highschool (percent over 25 who have completed high school), Divorce rate (of those who have married), and Politics (percent voting for Obama versus McCain in 2008, county-wide). To add complexity to the data set, we append 20 noise variables; i.e., variables that were generated from Gaussian distributions with means zero and variances comparable to that of either the latitude or longitude variables.

To visualize these data and assess varying structures in the data, we apply MDS as plotted in Figure 5A. To create this figure, all of the variables in the data set were weighted equally. Thus, the orientation we see of the data depends on both the real and noisy variable equally. A better orientation would isolate the important variables and down-weight those that are superfluous. For this reason, we develop V2PI for MDS.

3.4.2 Description of MDS. In a classical MDS scheme [3,22], the objective is to preserve pairwise distances between observations in low-dimensional representations of high-dimensional data. Using the same notation from the PCA example, we have a standardized data set $\mathbf{x}=[\mathbf{x}_1, \dots, \mathbf{x}_n]^T$ with n observations and $\mathbf{x}_i \in \mathbb{R}^p$ (for $i \in \{1, \dots, n\}$). We aim to estimate a low-dimensional version of \mathbf{x} that we denote by \mathbf{z} , where $\mathbf{z}=[\mathbf{z}_1, \dots, \mathbf{z}_n]^T$, $\mathbf{z}_i \in \mathbb{R}^q$ (for $i \in \{1, \dots, n\}$), and $q < p$. For the sake of visualization, $q=2$ and, for our above example, $p=30$. MDS solves for \mathbf{z} by minimizing the

absolute difference between pairwise distances of observations in \mathbf{x} and \mathbf{z} ,

$$\mathbf{z} = \min_{\mathbf{z}_1, \dots, \mathbf{z}_n} \sum_{i < j \leq n} \left| \|\mathbf{z}_i - \mathbf{z}_j\| - \delta(\mathbf{x}_i, \mathbf{x}_j) \right|, \quad (4)$$

where $\delta(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\|$, and $\|a - b\|$ is a predefined norm of the distance between points a and b . The right hand side of Equation (4) is typically referred to as a *stress* function, and the resolved minimum is called *the stress*. The norms used in Equation (4) will influence the MDS solution, if the distances themselves are sensitive to the norm under which they are computed. A common choice is the L_2 norm so that

$$\delta(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{d=1}^p (x_{id} - x_{jd})^2}, \quad (5)$$

where x_{id} and x_{jd} represent the d^{th} element in observations \mathbf{x}_i and \mathbf{x}_j respectively. This choice is arbitrary and can be adjusted easily to accommodate other norms.

Similar to PCA, MDS produces a spatialization of the data \mathbf{x} where relative distance between observations reflects their relative similarity. In fact, in the L_2 space, MDS will reproduce PCA visualizations. However, the explicit specification of a distance metric provides another means to parameterize feedback. For example, in the L_2 norm (Equation 5), all of the variables have equal importance or weight, even though we know that 20 variables are noise. Based on expert feedback, it makes sense to weight the variables in the distance metric so that only those that

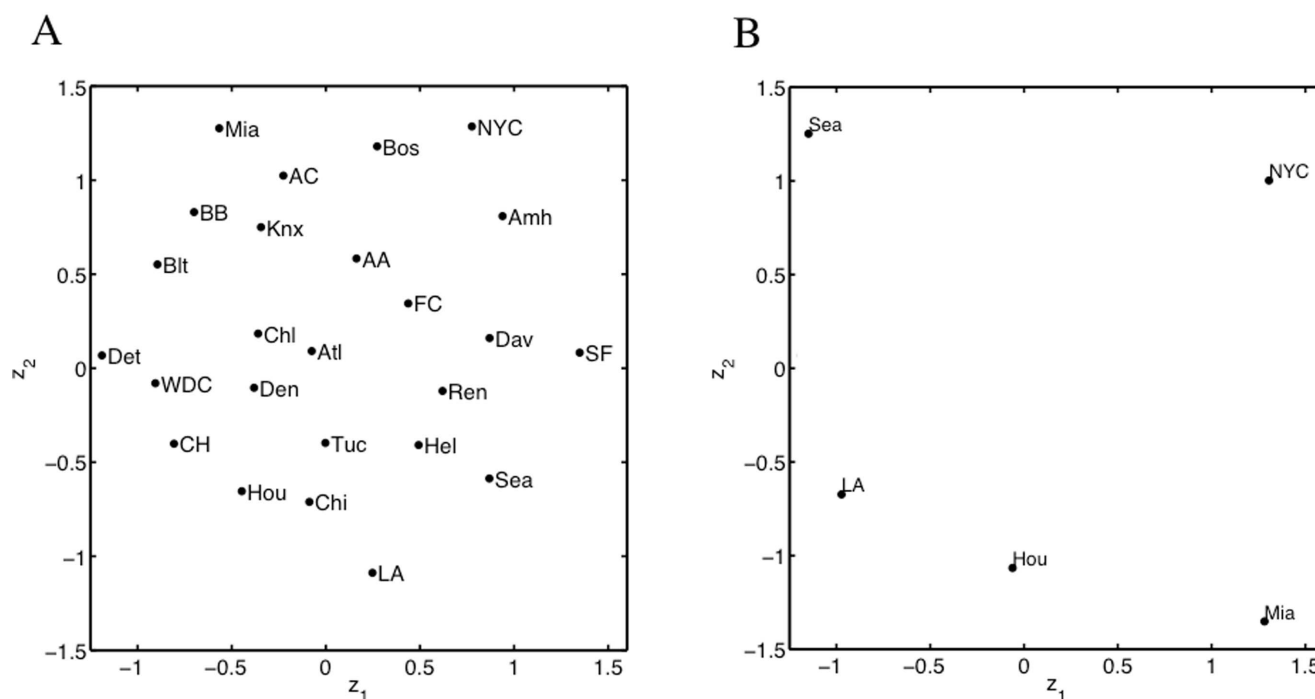


Figure 5. MDS view of the the city-data and an example of cognitive feedback. Figure A displays an Initial MDS view of the data set that describes 25 cities with 10 real variables and 20 noise variables. Figure B displays an example of cognitive feedback that arranges a set cities by relative geographic locations.

doi:10.1371/journal.pone.0050474.g005

are relevant influence the visualization. Next, we develop V2PI for MDS by reparameterizing the distance metric.

3.4.3 MDS with V2PI. To include expert judgments in MDS displays, we enable users to adjust (via cognitive feedback) a version of MDS known as Weighted Multi-Dimensional Scaling (WMDS) [23,24]. Just as MDS, WMDS minimizes the stress function in Equation (4) to find a solution for \mathbf{z} . However, WMDS replaces $\delta(\cdot)$ (the L_2 norm of two high-dimensional observations) with a weighted norm $\delta^w(\cdot)$,

$$\delta^w(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{d=1}^p w_d (x_{id} - x_{jd})^2},$$

where \mathbf{w} represents a pre-specified p -vector of dimension weights, $\mathbf{w} = \{w_1, \dots, w_p\}$, and $\sum_{d=1}^p w_d = 1$. Given \mathbf{w} , variables with large weights have more relevance than those with low weights in WMDS displays. Also, the MDS and WMDS solutions for \mathbf{z} are identical when $w_i = 1/p$ for each $i \in \{1, \dots, p\}$.

Now, using the data from Section 3.4.1 and the bi-directional pipeline (Figure 3) as a guide for our discussion, we develop V2PI for MDS based on the WMDS machinery. We start by applying WMDS with $w_i = 1/p$ for **Steps 1** and **2** of the bi-directional pipeline. For **Step 3**, experts may reconfigure three or more observations to reflect a conjecture about the data and communicate cognitive feedback F_c . For example, since the data describe 25 cities, it might be helpful to visualize how the cities distribute geographically across the United States. However, we can see from Figure 5A that the cities are geographically misplaced; e.g., no matter how we rotate the display, Seattle, Miami, San Francisco and Houston will not orient geographically. Thus, as cognitive feedback F_c , we enable MDS users to rearrange the locations of three or more cities to create V' . Figure 5B shows a possible orientation of five cities, Seattle (Sea), Los Angeles (LA), Houston (Hou), Miami (Mia), and New York City (NYC). Note that, similar, to PCA, there are a variety of surface-level interactions (which may eventually have a parametric interpretation) that users could perform with MDS visualizations. For this paper, we selected one.

The reason cities that do not map geographically is that the information in the variables, Latitude and Longitude, is masked by the remaining variables (both real and noise). Thus, for **Step 4** of the bi-directional pipeline, we parameterize F_c in the form of a weight vector that will up-weight the variables which seem to best explain V' and down-weight those that do not. Let A represent the set of k observations that were adjusted so that matrices \mathbf{x}^A and \mathbf{z}^A include only the high-dimensional and adjusted low-dimensional coordinates of the selected observations. To estimate new weights F_p , we solve the inverse problem; we solve for the weights that minimize the stress function based only data \mathbf{x}^A and \mathbf{z}^A . Explicitly, $F_p = \mathbf{w}$, where

$$F_p = \min_{w_1, \dots, w_p} \sum_{i < j \leq K} \|\mathbf{z}_i^A - \mathbf{z}_j^A\| - \delta^w(\mathbf{x}_i^A, \mathbf{x}_j^A). \quad (6)$$

The solution F_p is found easily using a gradient search method [25] with the constraint $\sum_{d=1}^p w_d = 1$. In our example, with cognitive feedback displayed in Figure 5B, the solution F_p weights Latitude and Longitude by 0.47 and 0.52, respectively. The total weight of the remaining variables equals 0.01. Note, based on F_p , users may learn the variables that define the structure they find to

be important. In this case, the users “learn” that the Latitude and Longitude explain their arrangement the observations because they have the largest weights. Although, we advocate the suppression of parametric information to users, weights have intuitive scales that are easy to interpret. That said, in an extremely high dimensional examples, creative, additional VA methods would be needed to provide this parametric information to users.

To assess how the remaining data spatialize given cognitive and parametric feedback, we technically apply a weighted average as described in Section 2.2.3 so that

$$\mathbf{w}^F = \rho F_p + (1 - \rho) \mathbf{w},$$

but we set $\rho = 1$ for this application of V2PI. In the standard MDS or WMDS procedures, the weight vector is pre-specified and set independently of the data, thus we do the same with V2PI. However, now, the weight vector is set according to user feedback. Subsequently, new data visualizations are created with the WMDS machinery and weight vector \mathbf{w}^F . Figure 6 includes a new visualization of the data. Since Latitude and Longitude define the geographic locations of cities and we want to demonstrate the success of MDS with V2PI, we re-scaled and rotated the updated low-dimensional coordinates so that we could overlay them on a US map. On this map, we also include the true city coordinates. The user-guided visualization approximates the true map fairly well.

From our exploration of the data with V2PI and MDS, users a) visualize how the cities in the data set distribute across the United States from Figure 6 and b) learn from F_p that Latitude and Longitude are the primary variables that explain the visual differences between the cities; e.g., Seattle and Miami are the furthest cities apart in Figure 6 because they differ the most in Latitude and Longitude. The data exploration could stop here, if users wanted. Or, users may reiterate the bi-directional visualization pipeline and inject more cognitive feedback to assess the data from another perspective. To show the latter is possible, we continue with the data exploration using V2PI in the next section.

3.4.4 Continuation of MDS Data Exploration. Looking at Figures 5 and 6, we see that the data set includes major cities and college-towns. Suppose a user is unable to classify all of the observations (only a small set) and the user wants to learn which variables differentiate major cities from college towns. Figures 5A and 6 (two MDS visualizations of the data), do not help the user. Thus, the users apply V2PI again.

For cognitive feedback F_c , users move the cities about which the classification is known into two separate groups. Specifically, three college-towns, Blacksburg (BB), Davis (Dav), and Fort Collins (FC), are placed away from two cities, New York City (NYC) and Washington D.C. (WDC), to create V' (Figure 7A). As previously shown, the methods are in place to learn weights F_p (the parameterized form F_c), set \mathbf{w}^F to F_p , and create a new visualization (Figure 7B). According to F_p , the selected observations differ most by Politics, Highschool, Age, and Population Density with weights 0.62, 0.22, 0.05, and 0.03, respectively. Also, based on proximity in Figure 7B, we see that Amherst (Amh), Ann Arbor (AA), Bloomington (Blt), and Chapel Hill (CH), are more similar to the selected college-towns (in the up-weighted variables) than the cities; and conversely, Boston (Bos), Chicago (Chi), Denver (Den), Detroit (Det), and San Francisco (SF), are more similar to the cities than the college-towns.

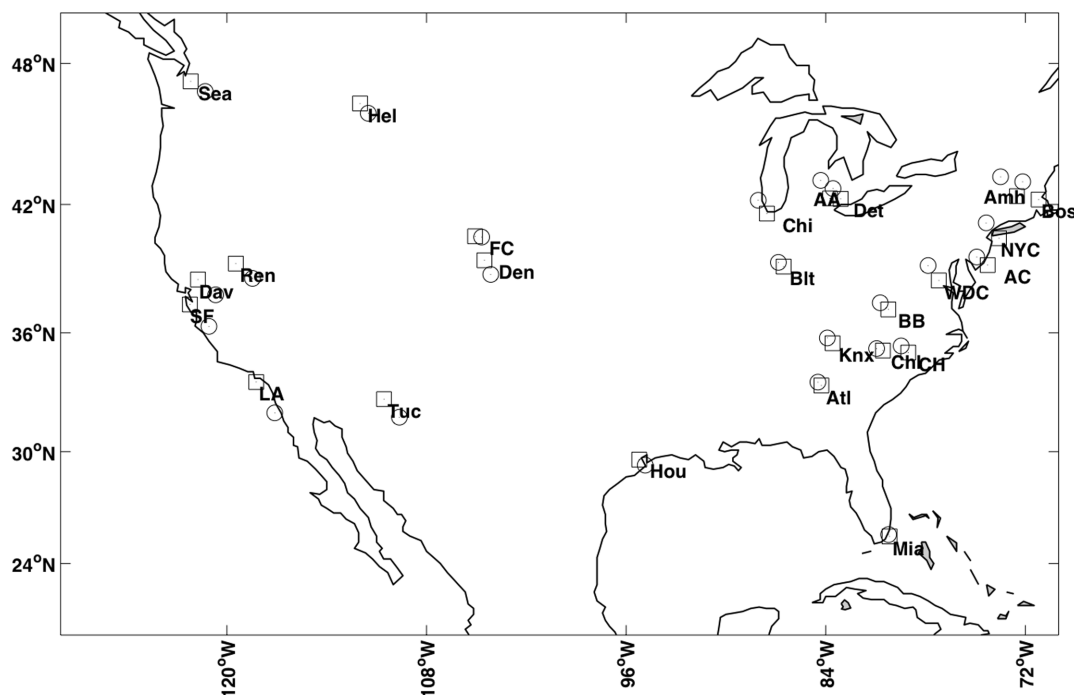


Figure 6. A visualization of the city-data that was updated by a parametric version of the cognitive feedback plotted in Figure 5B. The updated locations of the cities were stretched and rotated to overlay on a map of the United States. The symbols \square and \circ mark true and projected city coordinates by WMDS- V2PI. The estimated and true city coordinates are close.
doi:10.1371/journal.pone.0050474.g006

Discussion

We applied V2PI using two common data visualization methods. In each example, constraints in the mathematical

characterization of the data limited the utility of initial data displays; i.e., Figures 4A and 5A did not reveal expected nor meaningful structure. In turn, we included users in the visualization pipeline via V2PI to guide the mathematics and obtain

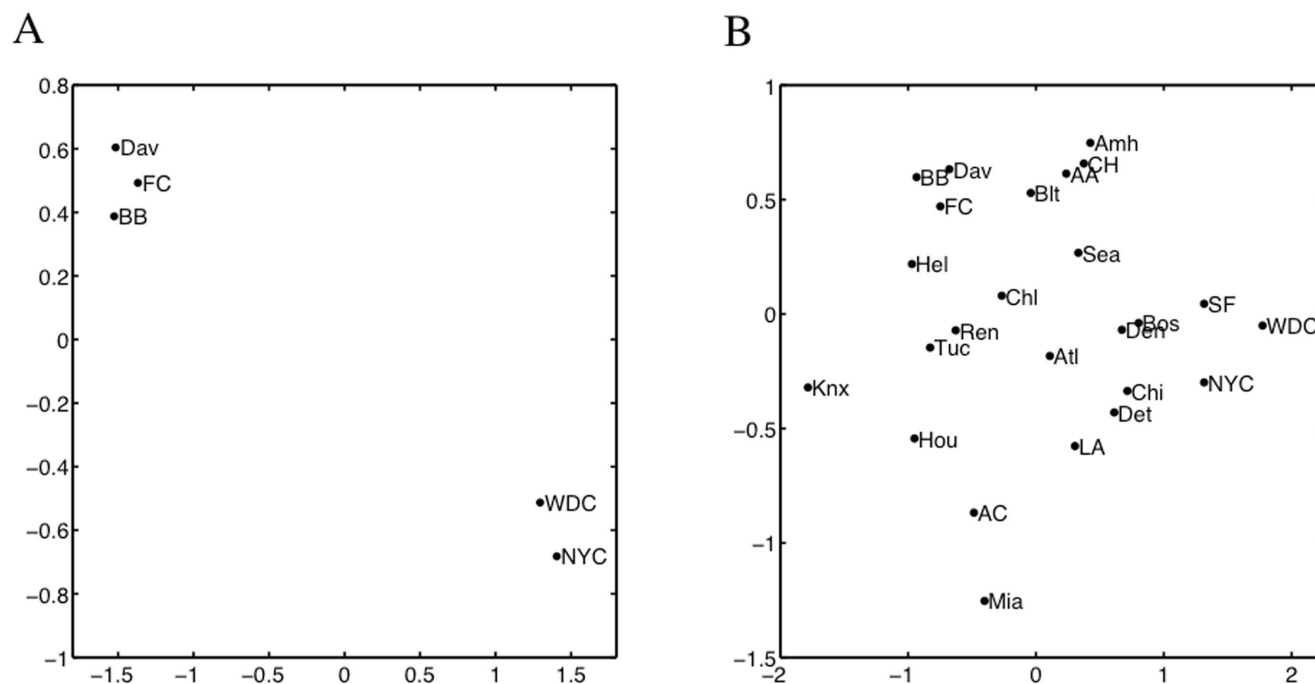


Figure 7. New cognitive feedback and updated view of city-data. Figure A plots another example of cognitive feedback that groups college towns separately from large cities. Figure B plots an updated visualization of the data that accounted for the feedback in Figure A.
doi:10.1371/journal.pone.0050474.g007

visualizations worth assessing. The case studies provided successful examples and avoided some practical challenges that we discuss here.

V2PI does not guarantee the display of obvious data structure; V2PI only guarantees to portray user intent-as interpreted parametrically within the constraints of the display-generating model. For example, in the case studies, V2PI guaranteed new spatializations that reflected the users' judgements about the observations' pairwise relationships, as defined by PCA and MDS. The improved depictions of the pairwise relationships were estimated by one, updated linear projection (by either PCA or MDS) of the data. Given different visualization methods, the pairwise relationships might have updated differently. For example, V2PI could be developed for other visualization methods, including, Isomap, Generative Topographical Models, and Mixture PCA [26–28]. Such approaches characterize data spatially using non-linear methods and/or multiple visualizations. Had V2PI been in place with these methods, the updated displays of data in Figures 4C, 4D, 6, and 7B might have configured the observations differently.

It is important that an appropriate method is applied to assess data visually. V2PI, in its current form, only enables parameter adjustments within the chosen methods, not adjustments to the methods themselves. VA tools that enable users to switch the underlying analytical methods of visualization could be useful, as the selection depends upon both characteristics of the data and the analytical goal of the data exploration. In the case studies, there was little to no difference between PCA or MDS to visualize the datasets. We developed them to demonstrate differences in how we can conceptualize and parameterize feedback. Had either dataset within the case studies included outliers or non-linear relationships between observations neither PCA nor MDS (based on the L_2 norm) would have been appropriate; non-linear methods, such as Isomap [26] or Generative Topographical Mapping [27] might be better. Also, there are several visualization methods that do not use geographic metaphors to display information. For example, cluster algorithms or network models may plot dendrograms or directed graphs to group and link one or more observations together. Albeit, clusters were revealed in the first, PCA case study, but PCA is not formally a cluster-discovering algorithm; the cluster assignments were up to the user (which has advantages and disadvantages). If a user wants to formally classify observations, an appropriate analytical method should be applied and V2PI can be developed accordingly.

Crucially, the selected analytical method determines both the ways by which users can communicate cognitive feedback and how it is parameterized. This was discussed in Section 2.2. In the case studies, we presented only one form of cognitive feedback per analytical method; users adjusted the locations of either two or more observations. However, there are multiple forms of cognitive feedback that are applicable to data spatializations, including filtering, querying, and annotating, that can be parameterized. Future work of this research includes the implementation of user studies to learn the various forms in which it is natural for users to convey cognitive feedback based on a variety of visualizations.

In such user studies, we would also assess how analysts learn to use VA tools with V2PI and quantify what they gain from V2PI. Different analytical methods and datasets may result in visualizations that vary in difficulty to interpret. Only once users understand the meaning of the displays, can they effectively inject feedback and make sense of data. Thus, V2PI is most advantageous when users can learn how to interpret displays and interact with them more efficiently than understand the display-generating parameters. For example, in the first case study with PCA, it is

reasonable to argue that users can assess and compare the relative differences between observations in visualizations with less effort than interpret the meaning of variance matrices, eigenvalues, and eigenvectors. Also, with V2PI, users can compare observations with PCA from varying perspectives to discover multiple structures or relationships in data easier than without V2PI. User studies that evaluate how well analysts understand PCA visualizations and compare what analysts learn using PCA with and without V2PI would likely support this argument. We envision a study that asks several questions about dataset(s) that analysts would be challenged to answer using what they know and VA tools with and without V2PI capabilities. The answers to the questions and the time it takes to answer the questions would illuminate the ease at which the analysts interpret the visualizations and the utility of the V2PI.

As with any user study, the dataset(s) that analysts are requested to assess is an important experimental-design element and may impact study results for several reasons. For example, datasets about which some users have prior knowledge and others do not will confound study findings. To evaluate V2PI, the size of datasets (in both the number of rows and columns) is also important to consider. Large datasets may impact the interpretability of some visualizations and computation time. In Figure 4, the dataset is small enough so that the points (i.e., observations) are distinguishable; distance between many observations was clear enough to inject feedback. Given millions of overlapping observations, this might not be the case. Also, with V2PI, real-time visualization updates enable users to explore data in parallel with them learning or thinking about the data. Yet, V2PI, as any analytical method, is limited by computational feasibility and efficiency. Thus, it is important to select datasets for studies that meet the constraints of the analytical method. Or, conversely, select/develop analytical and updating methods that can scale with data size. Both PCA and MDS (for certain distance metrics) are scalable.

Conclusion

In this paper, we discussed two fundamental concepts: the bi-directional visualization pipeline and V2PI. When we combine the two, we have a visualization scheme that enables experts to explore data from multiple perspectives without understanding the display-generating models. Since users do not need to understand the mathematical underpinnings of visualizations, they are free to build upon their knowledge base and merge their expertise with the information in data instantly. That is, they may have an opportunity to learn and interact with a dataset directly in its visual domain—the domain in which experts host their expertise and intuition.

An important feature of the bi-directional pipeline is that users receive and distribute information, thus both expert judgement and standard datasets are valid components of quantitative analyses (that underlie data visualizations). The use of each component is not particularly novel when analyses are constructed within the Bayesian paradigm. Bayesian models combine prior distributions that may represent subjective, expert-driven information, with likelihoods to formulate inferences. However, the bi-directional pipeline 1) does not require formal probabilistic specifications to operate; 2) enables experts to communicate their judgements via data visualizations; and 3) allows experts to inject their judgements during multiple stages of data analyses. Experts have multiple opportunities to recall, include, and reflect upon their judgements in analyses by adjusting visualizations at each iteration of the bi-directional pipeline.

We exemplified the use of the bi-directional visualization pipeline and V2PI within two case studies. For each, a projection method was used to spatialize data in two dimensions and we described a unique approach to implement V2PI. The approaches differed due to the subtle differences in the projection methods. For visualizations that do not rely on linear projections, the bi-directional visualization pipeline and V2PI may still apply. However, V2PI has practical limitations that were discussed when we reflect on the case studies, including the selection of visualization methods that are appropriate for both the data and expert, the necessary learning curve for using V2PI, and computational feasibility. Each limitation is addressable with careful thought and flexible VA tools.

That said, successful interactions with data via visualizations rely upon the development of VA tools that support V2PI. The

best VA tools are intuitive and accessible to users with varying levels of expertise. In this paper, we did not mention aesthetic aspects of VA tools that need to be considered for human cognitive purposes. Rather, we discussed the analytical mechanics needed in tools that enable V2PI.

Supporting Information

File S1 Parameterizing feedback for PCA.

(PDF)

Author Contributions

Conceived and designed the experiments: SL LH DM AE CN. Analyzed the data: SL LH DM AE CN. Contributed reagents/materials/analysis tools: SL LH DM AE CN. Wrote the paper: SL LH DM AE CN.

References

- Andrews NO, Fox EA (2007) Recent developments in document clustering. Technical report, Computer Science, Virginia Tech.
- Jolliffe I (2005) Principal Component Analysis. New York: Springer-Verlag.
- Kruskal JB, Wish M (1978) Multidimensional scaling. Sage University Paper series on Quantitative Application in the Social Sciences 48: 07–011.
- Keim DA (2002) Information visualization and visual data mining. IEEE Transactions On Visualizations and Computer Graphics 7: 100–107.
- Keim D, Mansmann F, Schneidewind J, Thomas J, Ziegler H (2008) Visual analytics: Scope and challenges. In: Simoff S, Bohlen M, Mazeika A, editors, Visual Data Mining: Theory, Techniques and Tools for Visual Analytics, Springer-Verlag, Berlin. pp. 767–90.
- Icke I, Sklar E (2009) Visual analytics: A multifaceted overview. Technical report, City University of New York.
- Pike WA, Stasko J, Chang R, OConnell TA (2009) The science of interaction. Information Visualization 5: 78–99.
- Jeong DH, Ziemkiewicz C, Fisher B, Ribarsky W, Chang R (2009) ipca: An interactive system for pca-based visual analytics. Computer Graphics Forum 28: 767–774.
- Buja A, Swayne DF, Litman ML, Dean N, Hoffman H (1998) Xgvis: Interactive data visualization with multidimensional scaling. Journal of Computational and Graphical Statistics 5: 78–99.
- Endert A, Han C, Maiti D, House L, Leman S, et al. (2011) Observation-level interaction with statistical models for visual analytics. In: Visual Analytics Science and Technology (VAST) 2011 IEEE Conference. pp. 121–130.
- Lederberg J (1989) Excitement and Fascination of Science, Twelve-Step Process for Scientific Experiments: Epicycles of Scientific Discovery. Palo Alto, California: Annual Reviews, Inc.
- Thomas J, Cook K (2005) Illuminating the Path. National Visualizations and Analytics Center.
- Pirolli P, Card S (2005) Sensemaking processes of intelligence analysts and possible leverage points as identified through cognitive task analysis. Proceedings of the 2005 International Conference on Intelligence Analysis.
- Card S, Mackinlay J, Shneiderman B (1999) Information Visualization: Using Vision to Think. San Francisco: Morgan Kaufmann.
- Pak Chung W, Hetzler B, Posse C, Whiting M, Havre S, et al. (2004) In-spire infovis 2004 contest entry. Proceedings of the IEEE Symposium on Information Visualization (INFOVIS'04).
- Norman DA (1990) The Design of Everyday Things. Doubleday Books, New York.
- Abowd G, Beale R (1991) Users, systems, and interfaces: Unifying framework for interaction. In: Human Computer Interaction HCI91: People and Computers. Cambridge University Press, Cambridge, volume 6, pp. 73–87.
- Myers B, Hudson SE, Pausch R (2000) Past, present, and future of user interface software tools. ACM Trans Comput-Hum Interact 7: 3–28.
- House L, Leman SC, Han C (2012) Bayesian visual analytics: Bava. Technical report, Department of Statistics, Virginia Tech.
- Koren Y, Carmel L (2003) Visualization of labeled data using linear transformations. In: Proceedings of the Ninth Annual IEEE Conference on Information Visualization. Washington, DC, USA: IEEE Computer Society, INFOVIS'03, pp. 121–128. URL <http://dl.acm.org/citation.cfm?id=1947368.1947392>.
- Hastie T, Tibshirani R, Friedman J (2008) Elements of Statistical Learning: Data mining, Inference and Prediction (2nd Edition). Springer-Verlag.
- Torgerson WS (1958) Theory and Methods of Scaling. New York: John Wiley.
- Carroll JD, Chang JJ (1970) Analysis of individual differences in multidimensional scaling via an n-way generalization of eckart-young decomposition. Psychometrika 35: 238–319.
- Schiffman SS, Reynolds ML, Young FW (1981) Introduction to Multidimensional Scaling: Theory, Methods, and Applications. New York: Academic Press.
- Mordecia A (1976) Nonlinear Programming: Analysis and Methods. New Jersey: Prentice-Hall.
- Tenenbaum JB, V de Silva JCL (2000) A global geometric framework for nonlinear dimensionality reduction. Science 290: 2319–2323.
- Bishop CM, Svensén M, Williams CKI (1998) GTM: the generative topographic mapping. Neural Computation 10: 215–234.
- Tipping ME, Bishop CM (1999) Mixtures of probabilistic principal component analyzers. Neural Computation 11: 443–482.