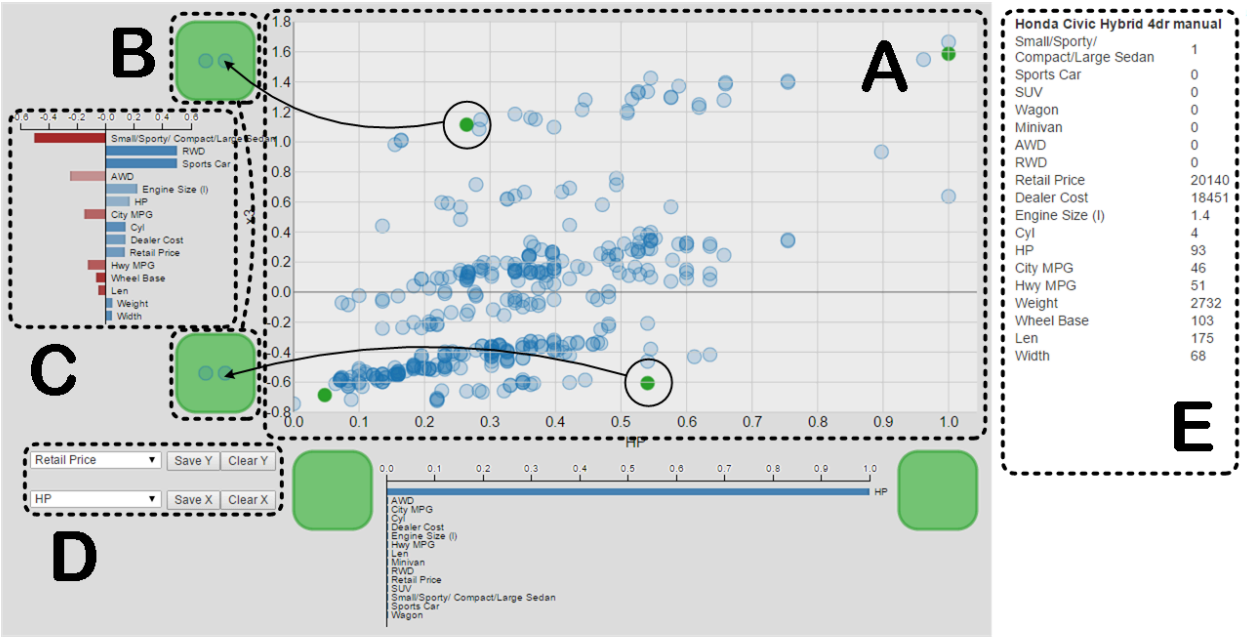
InterAxis: Steering Scatterplot Axes via Observation-Level Interaction

Hannah Kim, Jaegul Choo, Haesun Park, and Alex Endert



Honda Civic Hybrid 4dr manual	
Small/Sporty/ Compact/Large Sedan	1
Sports Car	0
SUV	0
Wagon	0
Minivan	0
AWD	0
RWD	0
Retail Price	20140
Dealer Cost	18451
Engine Size (l)	1.4
Cyl	4
HP	93
City MPG	46
Hwy MPG	51
Weight	2732
Wheel Base	103
Len	175
Width	68

Fig. 1. An overview of the proposed visual analytics system, InterAxis, showing a car dataset, which includes 387 data items with 18 attributes. The proposed system contains three panels: (A) the scatterplot view to provide a two-dimensional overview of data, (B-D) the axis interaction panel to support the proposed interaction capabilities, and (E) the data detail view to show the original high-dimensional information of the data items of interest. The axis interaction panel (B-D) consists of (B) two drop zones (the high-end and the low-end of each axis), which a user drags data points into in order to steer the axis, (C) an interactive bar chart, and a sub-panel containing buttons to save the current axis for future use (D, middle) or to clear the data points currently assigned to the axis (D, right) and a combo box to change the axis back to one among the original features or the previously created axes via our interaction (D, left).

Abstract—Scatterplots are effective visualization techniques for multidimensional data that use two (or three) axes to visualize data items as a point at its corresponding x and y Cartesian coordinates. Typically, each axis is bound to a single data attribute. Interactive exploration occurs by changing the data attributes bound to each of these axes. In the case of using scatterplots to visualize the outputs of dimension reduction techniques, the x and y axes are combinations of the true, high-dimensional data. For these spatializations, the axes present usability challenges in terms of *interpretability* and *interactivity*. That is, understanding the axes and interacting with them to make adjustments can be challenging. In this paper, we present InterAxis, a visual analytics technique to properly interpret, define, and change an axis in a user-driven manner. Users are given the ability to define and modify axes by dragging data items to either side of the x or y axes, from which the system computes a linear combination of data attributes and binds it to the axis. Further, users can directly tune the positive and negative contribution to these complex axes by using the visualization of data attributes that correspond to each axis. We describe the details of our technique and demonstrate the intended usage through two scenarios.

Index Terms—Scatterplots, user interaction, model steering

1 Introduction

Scatterplots are commonly utilized in visualizing relationships between two individual data attributes [13]. The use of two orthogonal axes mapped to data attributes produces a Cartesian space where data objects can be charted. A basic strategy to form these axes in multidimensional data visualization is to assign each axis an individual feature or dimension originally given in a dataset. For example, plotting temperature over time on the y and x axes, respectively, generates

- *Hannah Kim is with Georgia Institute of Technology. E-mail: hannahkim@gatech.edu.*
- *Jaegul Choo, the corresponding author, is with Korea University. E-mail: jchoo@korea.ac.kr.*
- *Haesun Park is with Georgia Institute of Technology. E-mail: hpark@cc.gatech.edu.*
- *Alex Endert is with Georgia Institute of Technology. E-mail: endert@gatech.edu.*

Manuscript received 31 Mar. 2015; accepted 1 Aug. 2015; date of publication xx Aug. 2015; date of current version 25 Oct. 2015.

For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

a chart that can be used for understanding the relationship between these two data attributes. However, this has a severe scalability issue because two-dimensional (2D) scatterplots can represent only two features out of many at any given point of time.

Instead, an alternative strategy that better handles this scalability issue is dimension reduction, which involves multiple original features to represent each axis. Dimension reduction [21] is a popular technique used to transform high-dimensional data into lower-dimensional views (typically, 2D scatterplots). While a variety of approaches exist, their fundamental functionality is similar: to solve for distances between data points in a lower-dimensional space that closely represents the true distances between the points in a high-dimensional space. This is carried out by variations in solving for distance metrics from the data.

In the visual and perceptual understanding of a scatterplot, the *interpretation of its axes* plays a crucial role. That is, understanding what it means to have large/small values along the x or y axis significantly helps the users' reasoning process about why the relationships among data items are close/remote in a scatterplot. In the case of traditional scatterplots where each axis is directly mapped to a particular data attribute (without any dimension reduction), this process is straightforward. However, this is not often the case when it comes to the axis of a 2D scatterplot generated by dimension reduction. One of the primary reasons is that only a limited set of dimension reduction methods provide the interpretability of the axes of a scatterplot. Such methods include traditional methods such as principal component analysis (PCA) [27] and linear discriminant analysis [23], which form an axis (or a reduced dimension) explicitly as a linear combination of the original data attributes. Through this linear combination representation of the original attributes, one can interpret the contribution of each original attribute to the axis. On the other hand, many other dimension reduction methods form each axis implicitly in terms of the original attributes, and thus they do not provide users with its clear meaning. Most advanced non-linear dimension reduction methods such as manifold learning [33] correspond to this case. Even worse, in some other popular methods such as multidimensional scaling (MDS) [31] and force-directed graph layout [22], these are rotation invariant, which means that the axis is not defined at all. Thus, communicating with users about the meaning of the axes resulting from dimension reduction techniques is an open challenge.

Another issue with the scatterplot generated by dimension reduction lies in the lack of *interactivity*. Forming the axes via dimension reduction does not typically allow human intervention. In other words, most of the dimension reduction methods are performed in a fully automated manner on the basis of their own pre-defined mathematical criteria, and thus, diverse user needs and task goals are not considered in this process. For instance, the PCA criterion, which maximally preserves the total variance of data, may not align well with the goal of a user's task. While MDS attempts to preserve all pairwise distances with equal weights, one may want to focus on a subset of data points, e.g., a local region in a scatterplot, at a time.

Motivated by these challenges, we propose a novel interactive knowledge specification method for multidimensional data visualization, which is an alternative to the purely automatic process of generating a scatterplot via dimension reduction. The proposed method interactively forms an axis, thereby generating a corresponding scatterplot in a user-driven manner. The key novelty of the proposed method lies in the direct and seamless incorporation of user-selected data items for characterizing the axis during the data exploration process. Our technique enables users to create and modify the axes by dragging data objects to the high and low locations on both the x and y axes. The proposed method defines the meaning of an axis accordingly in the form of a linear combination of original data features, similar to the output of linear dimension reduction methods. Such a user-driven linear combination of data attributes is visualized on each axis, showing the positive or negative contribution of each attribute to the axis. Finally, users can continually refine the axes by dragging additional data points to the axes, or by directly adjusting the contribution of the data attributes as part of the linear combination.

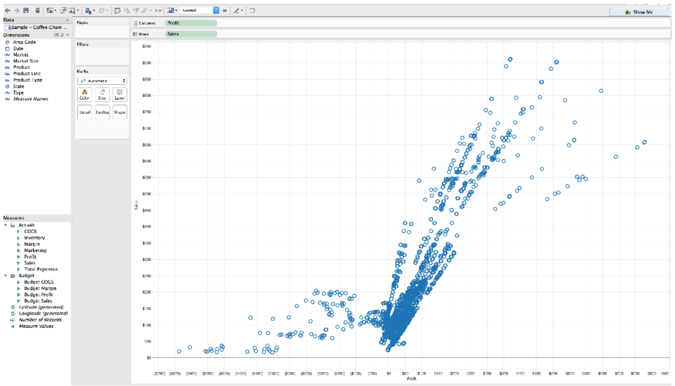


Fig. 2. A scatterplot generated by Tableau [41]. Users can interactively explore data by selecting and changing the bindings between data attributes and axes.

The primary contributions of this work include the following:

- a visual analytics technique for directly creating, modifying, and visualizing complicated axes formed by a linear combination of data attributes

- a user interaction technique enabling seamless interactivity via both data objects and data attributes to steer the meaning of the axes

- a visual analytics technique to help users discover and weigh data attributes

The rest of this paper is organized as follows: Section 2 discusses related work. Section 3 describes our proof-of-concept visual analytics system along with how the proposed interaction techniques are performed from the perspectives of both the front end and the back end, followed by a discussion about our design rationale. Section 4 presents several usage scenarios showcasing the advantages of the proposed interaction techniques. Section 5 presents in-depth discussions about the limitations of our interaction techniques as well as potential directions for improving them. Finally, Section 6 concludes the paper with some future work.

2 Related Work

In this section, we discuss previous work about the visualization applications of dimension reduction methods as well as user interactions with them.

2.1 Multiattribute Data Visualization

The design space for visualization techniques for representing multiattribute data is large [28]. For example, the existing techniques include iconic displays [6], transforming displays based on geometric characteristics [13], and stacked visual representations [32]. Among these many techniques, one commonly used technique is the scatterplot [12, 20, 45], owing to the visual simplicity and cultural familiarity of such charts [43]. Scatterplots (such as the one shown in Fig. 2) represent data on a Cartesian plane defined by the two graphical axes (the x and the y axes). Three-dimensional scatterplots are also an available option, but their use in information visualization is limited given the perceptual and visual challenges [38, 47]. Systems that enable users to generate scatterplots include Tableau [41], GGobi [40], Matlab [34], Spotfire [1], and Microsoft Excel [19]. One basic user interaction supported by scatterplots is to select and change the mapping of the axes to data attributes (Fig. 2).

As dataset complexities increase, often, the number of data attributes to select from increases as well. This causes situations where directly selecting one out of hundreds or thousands of data attributes can be less than optimal. As such, different types of techniques exist to show more combinations of data attributes simultaneously. For example, multiple scatterplots can be arranged into a single view called

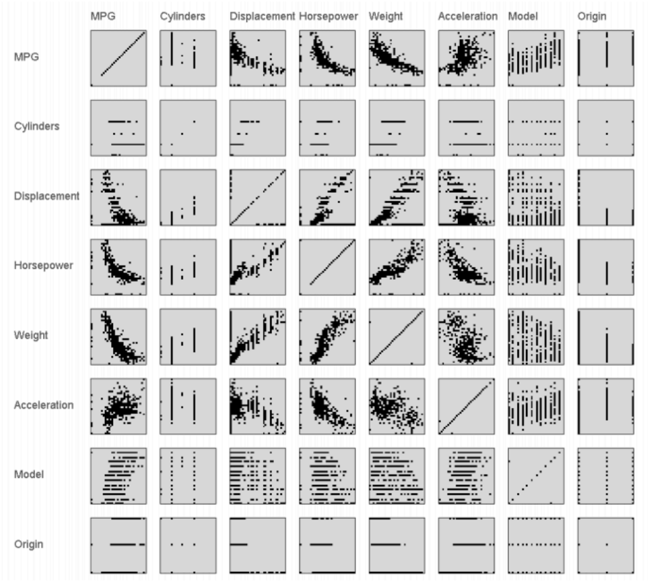


Fig. 3. A scatterplot matrix (adapted from [15]) showing all individual pairwise feature scatterplots of an 8-dimensional dataset.

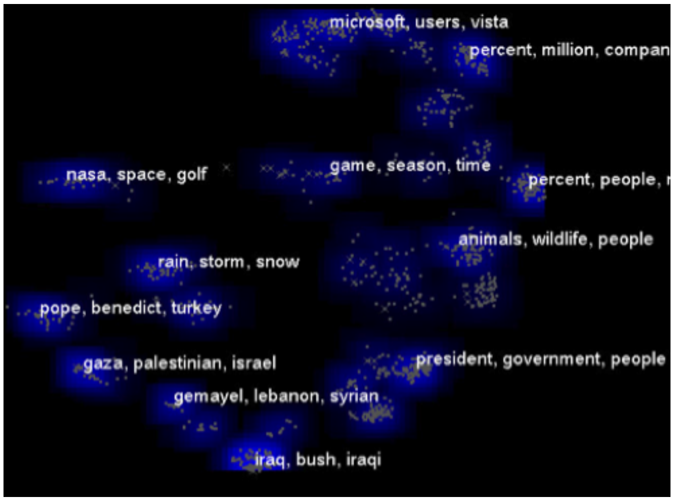


Fig. 4. A Galaxy View generated by IN-SPIRE [48] showing a scatterplot of documents (dots).

a scatterplot matrix [12]. A scatterplot matrix (such as the example shown in Fig. 3, adapted from [15]) binds data attributes to rows and columns so that each cell in the matrix can represent a single scatterplot. As such, users do not have to individually bind data attributes to the axes and interactively choose among the potentially large number of choices.

2.2 Applications of Dimension Reduction in Information Visualization

When using dimension reduction for visualization purposes, the goal is to provide a low-dimensional view, typically a 2D scatterplot, in a manner that the original high-dimensional distances between data points are maximally preserved in the resulting 2D views. These views often show spatial clusters or groups of data representing coherent contents. The widely used dimension reduction methods used for visualization include PCA [27], MDS [31], self-organizing map (SOM) [29], and generative topographic mapping (GTM) [3]. Recently, t-distributed stochastic neighbor embedding [46] has been proposed as a dimension reduction method, which is particularly suitable for generating 2D scatterplots that can reveal meaningful insights about data such as clusters and outliers.

To date, these methods have been actively adopted in visual analytics systems. For example, IN-SPIRE [48], a well-known visual analytics system for document analysis, provides a Galaxy View (as shown in Fig. 4) that visualizes text corpora spatially by showing the pairwise similarity between documents as their distance in a 2D space. As a result, groups and clusters emerge, which can be perceived as the sets of similar documents, based on the geographic "near=similar" metaphor [39]. More recently, a visual analytics system applicable to more general high-dimensional data types including documents and images has been proposed, allowing a user to explore the diverse aspects of data by applying various dimension reduction methods to generate different scatterplot visualizations [9].

Other kinds of high-dimensional data have also been visualized in the form of a scatterplot based on dimension reduction, including education performance data, census data [18], wine characteristics [5], facial images [8], and text documents [7].

2.3 Interactivity for Dimension Reduction in Information Visualization

In general, the axes created via dimension reduction techniques are defined by linear or non-linear combinations of original data dimensions. This complexity can lead to trust and interpretation challenges for domain experts exploring their data visually [10]. For example, users may question whether their interpretation of a pattern is trustworthy or if it is just an artifact of a dimension reduction technique. More fundamentally, using only two dimensions to represent considerably higher-dimensional data inevitably involves significant information loss and distortion. To overcome these issues, various user interactions have been employed in numerous visual analytics systems.

One approach to user interaction is via direct manipulation of dimension reduction model parameters. For example, Jeong *et al.* presented iPCA, a visual analytics application that visualizes high-dimensional data in a 2D scatterplot using PCA [26]. They utilize graphical controls (e.g., sliders) to enable users to directly manipulate the weight on the principal components used in PCA. As a result, the adjustments by the user generate a new projection (i.e., a new scatterplot). Similar interaction guidelines have been used by other applications, such as a text visualization system called STREAMIT [2].

A different set of techniques for incorporating user interactions into such visual analytics systems also exists. Semantic interaction techniques function by inferring model updates based on direct interactions performed in the visualization [16, 17]. For example, Endert *et al.* have shown how directly manipulating the position of points in a 2D scatterplot can be used for inferring the parameters of PCA, MDS, and GTM [18]. These inferences can also be used for exporting the specification of distance functions computed in the dimension reduction step so that they can be reused, shared, or simply saved [5].

Other than manipulating data items to interact with scatterplots, researchers have studied the interaction techniques that manipulate features or dimensions. Yi *et al.* have presented a technique called Dust & Magnet that allows users to additionally place features or dimensions on top of a scatterplot themselves to see which data items have large values of these features or dimensions [49]. For text analysis, the VIBE system allows users to perform similar interactions with keywords [35]. In addition, Turkay *et al.* proposed a technique using dual scatterplots one of which shows data items while the other shows features [44]. By providing brushing and linking as well as filtering operations on both data items and features in these dual scatterplots, users can check major patterns as well as outliers among data items and among features.

The technique proposed in this paper follows a similar idea of interacting with both data items and features, but the main novelty of the proposed technique against the existing work lies in the capability of directly defining and interpreting the axes of the 2D scatterplot by assigning the data items of our interest to the axes. In this respect, our work is related to PivotSlice, a technique recently proposed by Zhao *et al.* that allows faceted browsing of high-dimensional data [50], as it allows users to specify data attributes on axes of the scatterplot by directly dragging the attribute to the axis. However, our technique enables users to drag data objects (instead of data attributes) to the axis. Further, the proposed technique does not divide the scatterplot into a multifaceted view.

Furthermore, a technique called flexible linked axes [11] has a relationship with our work from a different aspect. That is, this technique is a different type of interaction that allows users to draw axes on a canvas, where scatterplots can be generated between any two neighboring axes. However, the main goal of this technique is fundamentally different from ours in that it attempts to flexibly coordinate and place multiple scatterplots on a large canvas, while our focus is on improving a single scatterplot for better supporting the interactive exploration of data based on a more sophisticated, user-driven axis specification. Further, Kondo and Collins have shown how directly interacting with visualizations can be used for revealing temporal trends and relationships between data items [30]. Their work allowed users to manipulate the position of data points in a scatterplot to reveal the temporal trends in data, again enabling interactions directly on the data items in a scatterplot to parameterize a data model.

3 Proposed Technique

To realize the proposed interaction technique, we built a proof-of-concept visual analytics system. In this section, we describe (1) the overall design of the proposed visual analytics system, (2) the proposed interaction to steer the axis in a user-driven manner, (3) the underlying mathematical details to support the proposed user interaction, (4) the design rationale, and (5) the implementation details of the proposed system.

3.1 System Design

As shown in Fig. 1 by using the well-known Car dataset, which consists of 387 data items with 18 attributes,[1] the proposed system mainly contains three panels: (1) the scatterplot view (Fig. 1(A)), (2) the axis interaction panel to support the proposed interaction capabilities (Fig. 1(B-D)), and the data detail view (Fig. 1(E)).

The user interaction technique presented in this paper fosters a visual data exploration process grounded in the principles of semantic interaction techniques [16, 17]. That is, the system interprets the analytical reasoning of exploratory user interactions to steer the underlying data model. The generic workflow supported by our user interaction technique is as follows:

1. The user observes two data points that define the difference between the two semantic groupings (e.g., “nice cars” and “bad cars”).
2. The user drags one data item to each side of the axis.
3. Interaxis computes the weighting of data attributes that supports these higher-level groupings (Eq. 1). The weights are displayed in the bar chart below the axis.
4. The scatterplot updates to reflect the newly defined axis, where data items are placed according to the similarity on either side of the axis (Eq. 2).
5. The user can refine the semantic grouping by adding/removing data points or directly modifying the weighting in the visualization below the axes.
6. The user can save the axis for future use and continue to explore the visualization iteratively by using the same interaction concept based on different semantic groupings.

The scatterplot view provides a 2D overview of the data. By default, the first and the second features of data, e.g., Retail Price and HP (Horsepower), are assigned to the x and the y axes, respectively, but this initial view can be set up by using a dimension reduction method such as PCA [27] to provide another starting point. Data points are represented as semi-transparent circles so that regions with overlapped data points can be highlighted. The scatterplot view supports zoom and pan via mouse wheel operations on a white space (to zoom on both axes simultaneously) or over a particular axis (to zoom only on this axis). Hovering over or clicking on a data point, one can check the full details (or the original high-dimensional information) of the data item in the data detail view (Fig. 1(E)).

The axis interaction panel consists of two drop zones (the high-end and the low-end of each axis), which the user drags data points into in order to steer the axis (Fig. 1(B)), an interactive bar chart (Fig. 1(C)), and a sub-panel (Fig. 1(D)) containing buttons to save the current axis for further use or to clear the data points currently assigned to the axis and a combo box to change the axis back to one among the original features or the previously defined axes. The bars in the interactive bar chart represent the contributions/weights of attributes to the corresponding axis. The longer the length of a bar is, the stronger its corresponding attribute contributes to the axis. The bars are color-coded by the signs of their weights: positive contributions in blue and negative contributions in red. Data points that are high on the positively weighted (blue-colored) attributes will be placed on the high-end side of the axis. Data points that are high on the negatively weighted attributes will be placed on the low-end side of the axis. For example, in Fig. 1(C), sedans tend to be on the left side of the scatterplot, while sports cars and cars with rear-wheel drive (RWD) tend to be on the right side. Positive and negative weights represent the magnitude and at which end of the axis the data points with those attributes will be placed.

3.2 Interactive Axis Steering

The proposed method provides two types of interactions: (1) data-level axis steering and (2) attribute-level axis manipulation. Data-level axis steering is prompted by dragging a data point from the scatterplot into the two drop zones at the high- and the low- end of the axis. Attribute-level axis manipulation is prompted by directly adjusting the bars in the interactive bar chart.

The main idea of the proposed interaction for steering the axis in a user-driven manner lies in an intuitive process of incorporating data items seamlessly while exploring data in a scatterplot. For example, when a user finds data points that he likes (or dislikes) in the scatterplot, he can drag them to the high-end (or the low-end) drop zone of an axis (Fig. 1(B)). Accordingly, a new axis is formed by reflecting these choices of data items, which will then update the scatterplot on the basis of the newly formed axis. The technical details about how we form a new axis will be described in the next section.

How the axis is formed from this process is summarized and visualized as a bar chart (Fig. 1(C)) so that a user can get an idea about how much a particular original feature or dimension is emphasized or de-emphasized. Given such a bar chart, a user can further refine the meaning of an axis by directly manipulating the length of each bar via drag-and-drop operations on the tip of the bar (attribute-level axis manipulation).

The entire interaction process can be dynamic and iterative. That is, a user can additionally assign new data items to an axis or remove data items that was already assigned to an axis. Furthermore, the above-described direct manipulation on the bar chart can be performed at any moment during such an interactive exploration of the bar chart. Finally, a user can save the current definition of an axis, and then it is registered as a new entry in the combo box (Fig. 1(D, left)) so that a user can later recover the axis to a previously saved one.

3.3 Underlying Techniques

In this section, we describe the underlying technique for the proposed user interaction of forming the axis via data items. For the sake of brevity, we consider only the x axis (the horizontal axis) in a scatterplot, but the following description can be generalized to the y axis in the same manner.

Data preprocessing. As will be discussed later, the underlying model to define the axis is based on a linear combination of the original dimensions. To this end, we adopt data preprocessing steps used

[1] http://www.idvbook.com/teaching-aid/data-sets/2004-cars-and-trucks-data/

Table 1. Notations used in this paper

Notation	Description
$n_{x,h}$	Number of data items assigned to the high-end of the x axis
$n_{x,l}$	Number of data items assigned to the low-end of the x axis
$n_{y,h}$	Number of data items assigned to the high-end of the y axis
$n_{y,l}$	Number of data items assigned to the low-end of the y axis
$A^{x,h} = \left\{a_i^{x,h}\right\}$	Set of data items assigned to the high-end of the x axis
$A^{x,l} = \left\{a_i^{x,l}\right\}$	Set of data items assigned to the low-end of the x axis
$A^{y,h} = \left\{a_i^{y,h}\right\}$	Set of data items assigned to the high-end of the y axis
$A^{y,l} = \left\{a_i^{y,l}\right\}$	Set of data items assigned to the low-end of the y axis
T_x, T_y	Linear transformation vectors for the x and the y axes, respectively

in linear regression models [14]. For a categorical variable with c different categories, we use dummy encoding, which converts it to a c-dimensional indicator vector where the value of each dimension is 1 if a data item is in the category of the corresponding dimension and 0 otherwise. Next, we scale and translate each dimension (including both indicator and numerical variables) so that its value is exactly in the range from 0 to 1.

Linear transformation. Assuming that such data preprocessing is done, we denote a set of high-dimensional vectors of data items that the user assigned (via a drag-and-drop) to the high-end of the x axis as $A^{x,h} = \left\{a_1^{x,h}, a_2^{x,h}, \cdots, a_{n_{x,h}}^{x,h}\right\}$ and a set of those that he dragged into the low-end side of the x axis as $A^{x,l} = \left\{a_1^{x,l}, a_2^{x,l}, \cdots, a_{n_{x,l}}^{x,l}\right\}$, where $n_{x,h}$ and $n_{x,l}$ represent the total number of the assigned points to the high-end and the low-end of the x axis, respectively. Now, we define the linear transformation vector for the x axis as follows:

$$T_x = \frac{1}{n_{x,h}} \sum_{i=1}^{n_{x,h}} a_i^{x,h} - \frac{1}{n_{x,l}} \sum_{i=1}^{n_{x,l}} a_i^{x,l}. \tag{1}$$

This is then further scaled to have a unit Euclidean norm.

One can define the linear transformation vector T_y for the y axis in the same manner. Every data item is mapped to the x axis (and the y axis) via the transformation T_x (and T_y). That is, the i-th data item whose high-dimensional vector is represented as a_i is mapped to a point in our 2D scatterplot so that its 2D coordinates are represented as follows:

$$\left((T_x)^T a_i, \, (T_y)^T a_i \right). \tag{2}$$

Owing to the easy interpretability of this linear model, one can understand the meaning of this transformation in a straightforward manner. That is, the resulting x axis basically emphasizes the features or dimensions that have large values on the high-dimensional vectors contained in $A^{x,h}$ but have low values on those in $A^{x,l}$. On the other hand, we de-emphasize the features that have low values on the vectors contained in $A^{x,h}$ but have high values on those in $A^{x,l}$. In this manner, as a data item has larger (or lower) values on these emphasized dimensions and lower (or higher) values on the de-emphasized dimensions, its x coordinate will have a higher (or lower) value, appearing more on the right (or left) side of the x axis. The notations used in this section are summarized in Table 1.

3.4 Design Rationale: Tradeoff between Explicit Parameter Control and Implicit Model Steering

In this section, we discuss the design rationale behind our visual analytic technique. The core focus of this technique is grounded in the tradeoff between a user interaction designed to require the adjustment of analytic model parameters directly, and the semantic interaction approaches that perform model steering through the inference of the user's actions [16, 17].

Considering the underlying linear model described in the previous section, an alternative design choice would be to let users manipulate the linear model (more exactly, linear combination coefficients) from scratch. In fact, this type of approach has been utilized in previous studies including iPCA [26], where users want to manipulate the linear combination coefficients from a PCA output.

However, this approach has several important drawbacks from a user's perspective, the explanation of which can be found in Shipman *et al.*'s discussion on the formality in computer-supported cooperative work [37]. In their work, Shipman *et al.* pointed out the discrepancy between the formality required by a computer and the formality that can be provided by a human. Such a discrepancy can be described from the two following aspects: The first issue is that users may have only *tacit* knowledge about what they want, which cannot be fully formalized at the beginning although a computer requires them to provide the full details right away, for example, understanding which data features to adjust (and by how much). The second issue originates from the tradeoff relationship between the amount of additional *formality* that the users have to provide and the additional *benefit* that they can get out of it. In other words, the capabilities requiring greater degrees of formality end up being less frequently used [37]

Therefore, with respect to interactive axis steering, the level of formality that a system requires enables the users to fully specify the axis as a linear combination representation in terms of all the features. However, users are not likely to have an exact idea of what the axis should be (*tacit* knowledge). Further, setting each of the linear combination coefficient values from scratch, particularly when the number of dimensions is large, can be a significant burden to users when trying to form the axis as they want (the *formality-benefit* tradeoff), which ends up making such a capability less useful.

On the other hand, our data-level interaction methodology nicely overcomes these challenges of *formality*. The fact that the system requires a full specification of a linear combination representation still holds. However, using our data-level interactions, users do not need to know the exact coefficient values in advance, but they only need to tell the system the data items that they place on the axes one by one to achieve the same capability. Through this iterative process, their tacit knowledge can be incrementally formalized to explore a scatterplot. Even though our interaction design contains direct manipulation capabilities of linear combination coefficients (through interactions with a bar chart), this is not a main process but an optional, fine-tuning one, allowing users to intervene when they have formalized their insights and questions later in the process.

3.5 Implementation

Our web-based visual analytics system is implemented using JavaScript, and the main visualization modules are built on the D3 toolkit [4], a widespread JavaScript information visualization library. Datasets are stored in a comma-separated values format, and the entire data are directly fetched into the visualization at the time the website is loaded. The implemented system can be accessed at `http://www.cc.gatech.edu/~hkim708/InterAxis`. Once a dataset is chosen among the several different ones that we prepared, a user can freely explore the data by using the proposed techniques.

In terms of computing performances, the proposed system can handle up to several thousands of data points with no noticeable delay. It is rather the front-end rendering module that suffers from the scalability issues since our underlying mathematical model described in Section 3.3 works efficiently with the time complexity of $O(n \times d + d \times \log(d))$, where n denotes the number of data points and d represents the number of features.

4 Usage Scenarios

In this section, we demonstrate the effectiveness of the proposed interaction technique in a multidimensional data exploration by using two

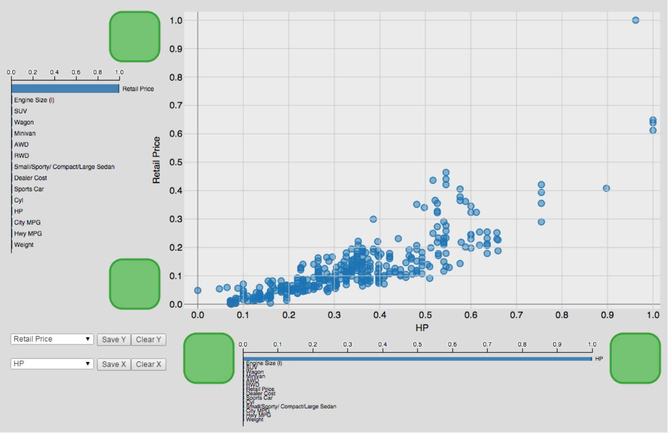


Fig. 5. An initial scatterplot showing HP by Retail Price.

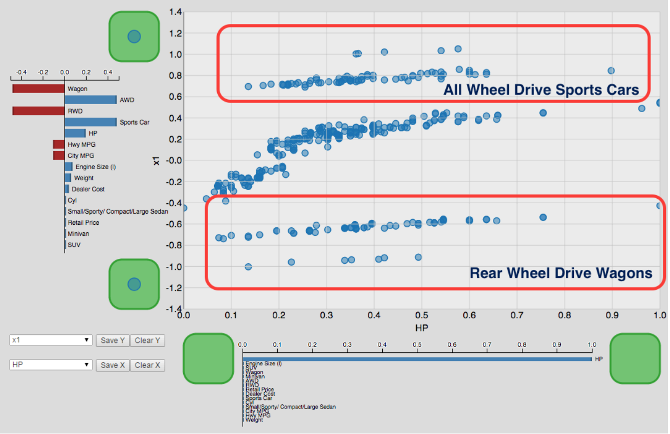


Fig. 6. A scatterplot obtained after assigning Subaru WRX STi on the high-end drop zone of the *y* axis and Pontiac Vibe on the low-end drop zone of the *y* axis.

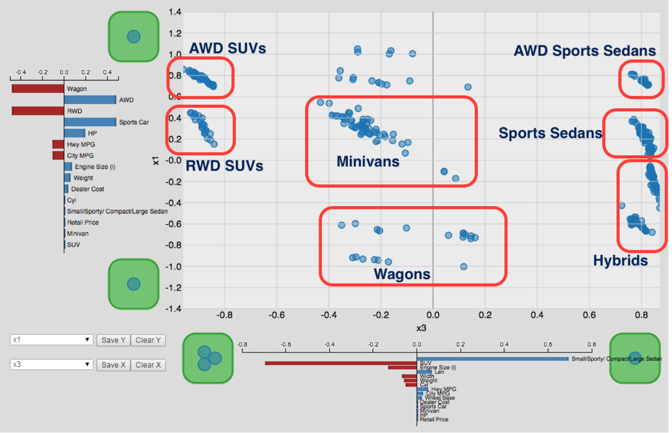


Fig. 7. A scatterplot obtained after assigning several SUV cars on the low-end and a sedan on the high-end drop zones of the *x* axis.

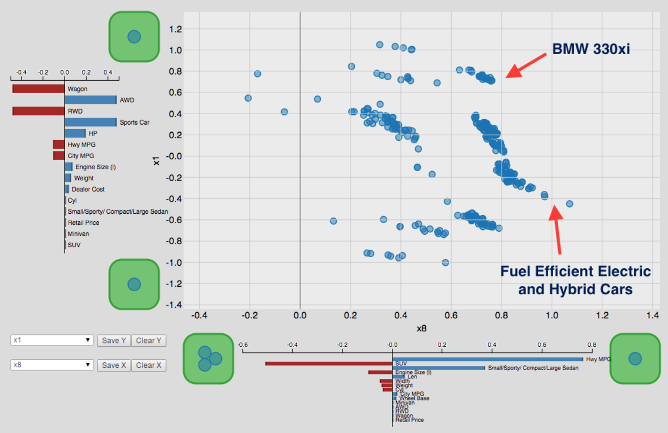


Fig. 8. A scatterplot obtained after increasing the weight on Hwy MPG on the *x* axis by dragging the bar directly to the right.

real-world datasets: the Car dataset and the Crime dataset.

4.1 Dataset Description

The car dataset provides specifications on new cars and trucks for the year 2004. The attributes of this dataset include vehicle categories, vehicle measurements, retail/dealer prices, and fuel efficiency. After removing data instances with missing values, we are left with 387 cars and 18 attributes.

The communities and crime dataset, which is available at `https://archive.ics.uci.edu/ml/datasets/Communities+and+Crime+Unnormalized`, aggregates socioeconomic data from the 1990 US Census, law enforcement data from the 1990 US Law Enforcement Management and Administrative Statistics (LEMAS) survey, and crime reporting data from the 1995 FBI Uniform Crime Reporting (UCR). After removing data instances with missing values, we are left with 1,901 communities and 112 attributes. We used 200 of the highest populated communities for the use case discussed in Section 4.3.

4.2 Car Dataset

To illustrate the functionality of our visual analytics technique, let us consider the following usage scenario. Amy is shopping for a vehicle to purchase and wants to understand and make an informed decision by using our interaction technique. Thus, she explores a dataset of 387 cars containing 18 attributes for each car. These attributes include categorical, nominal, and continuous variables such as Sports Car (a binary variable) and Retail Price (a continuous variable).

Her visual exploration of the data starts with an initial scatterplot that shows HP by Retail Price on the *x* and the *y* axes, respectively. From this view (shown in Fig. 5), she can see that there are cars that differ in terms of how much horsepower one can get for the amount of money that one has to pay. One of her friends drives a Subaru WRX STi, which she likes, so she decides to drag it to the high-end drop zone of the *y* axis. Her current car is a Pontiac Vibe, which she does not like, and thus, in order to contrast the two, she drags this one to the low-end drop zone of the *y* axis.

The proposed system computes the axis accordingly, presenting Amy with a new scatterplot (shown in Fig. 6). Upon exploring this view, she observes that her *y* axis reflects the difference between these two cars, such as whether or not a car has all-wheel drive (AWD), is a sports car, is not a wagon, and has high HP, which can be checked from the bar chart by summarizing these differences. The scatterplot reflects these criteria. The layer across the bottom consists of wagons, and the layer across the top consists of all sports cars, with the subset across the very top consisting of AWD sports cars.

However, she observes that some of the cars in this top half are SUVs (given that they have lots of HP and some of them are AWD). She drags three of these to the low-end drop zone of the *x* axis and takes one of the sedans and places it in the high-end drop zone. The resulting re-calculated view is shown in Fig. 7.

From this view, she can see that the *x* axis now contains small/compact/large sedans on the right side and SUVs on the left side. She browses the clusters revealed through this interaction to get a better understanding of what the visual groupings mean (as shown in Fig. 7). She sees that a part of her *x* axis is now defined by fuel economy (Hwy MPG). She had not considered this attribute of a car and decides that she wants to be somewhat economical with her choice. She increases the weight on the Hwy MPG attribute by directly dragging the blue bar further to the right.

As a result, as shown in Fig. 8, she sees that while there are very

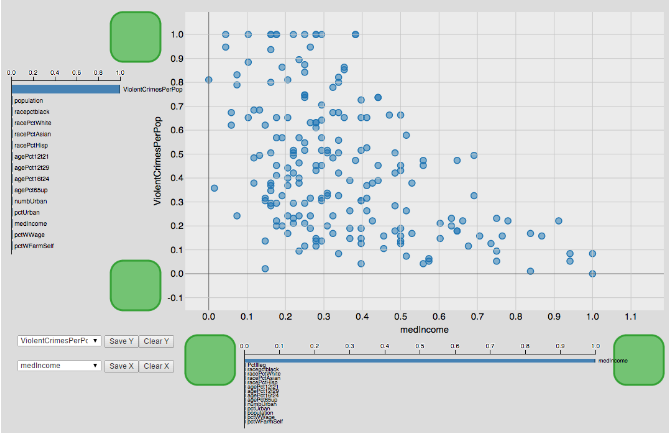


Fig. 9. An initial scatterplot showing ViolentCrimesPerPop (the total number of violent crimes per 100K population) by medIncome (median household income).

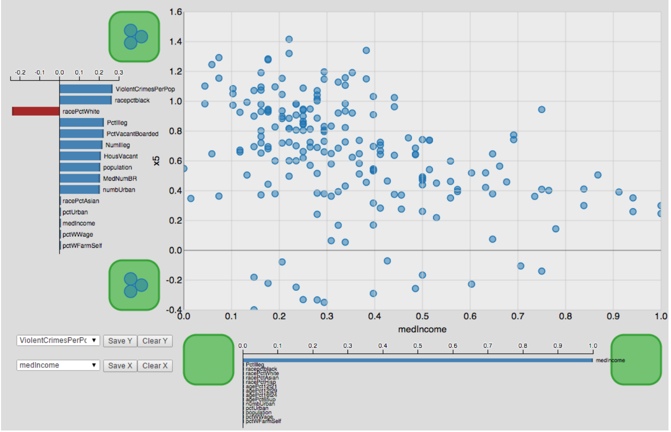


Fig. 10. A scatterplot obtained after assigning high-crime cities in the high-end drop zone of the *y* axis and low-crime cities in the low-end drop zone of the axis.

fuel-efficient cars to the right side of her *x* axis, they are on the lower part of her *y* axis (i.e., similar to the car that she does not like). She inspects them and finds cars such as the Toyota Prius and Honda Insight. Instead, she inspects the cluster on the top right, representing cars that are high on the *y* axis (i.e., similar to cars she likes), and on the right side of the *x* axis (sedans with good fuel economy). She finds the BMW 330xi from this cluster and heads to the dealership with a printout of the visualization and confidence in her decision.

4.3 Communities and Crime Dataset

Now, we will follow the case of a tourist who uses the proposed technique to examine a communities and crime dataset. This dataset contains socioeconomic, law enforcement, and crime data. This dataset is in contrast to the car dataset used in the previous usage scenario in that it contains more data attributes (112, as compared to 18 for the car dataset). As such, the task of our user is to discover the data attributes that help explain notions that she has about specific cities. For example, she has the cities that she has been to and enjoyed: Does the data support her preference? Can she discover what tradeoffs she is implicitly making that she is not aware of? Similarly, what about cities that she considers "safe" or "dangerous"?

She loads the application and starts by looking at the initial representation (shown in Fig. 9) visualizing a scatterplot of the median household income by the total number of violent crimes per 100K population on the *x* and *y* axes. She drags three of the high-crime cities to the top drop zone of the *y* axis, and three of the low-crime cities to the bottom drop zone. She refers to the top of her *y* axis as "dangerous cities", and the bottom as "safe". Her goal is to discover which data attributes describe these two notions about cities.

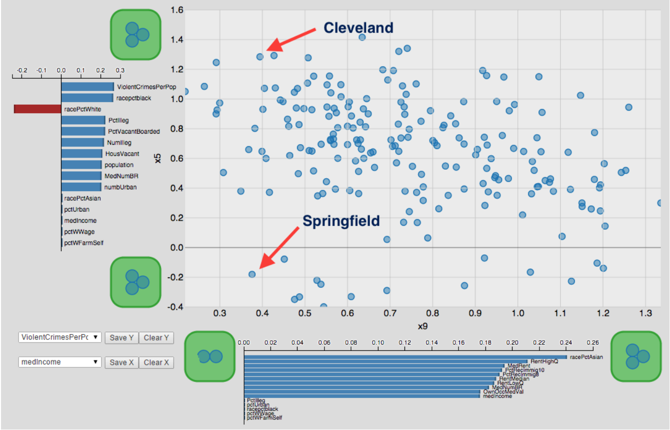


Fig. 11. A scatterplot obtained after assigning Santa Clara, San Jose, and Irvine to the high-end drop zone of the *x* axis and New Orleans and Green Bay to the low-end drop zone of the axis.

In Fig. 10, she observes the attributes that are shown to correspond to "dangerous cities" are shown on the *y* axis (e.g., racePctBlack, HouseVacant, etc.). The application sorts the list of attributes by magnitude, placing the most dominant attributes at the top. She discovers that the percentage of vacant houses helps describe her notion of these cities. Reflecting on her time spent in these cities, she is reminded of passing by many vacant houses.

Next, she decides to place some of the cities that she has been to and liked in the high-end drop zone of the *x* axis, and the cities that she did not like in the low-end drop zone of the axis. On the right (high-end), she places Santa Clara, San Jose, and Irvine. On the left (low-end), she places New Orleans and Green Bay. Fig. 11 shows the updated visualization based on this feedback.

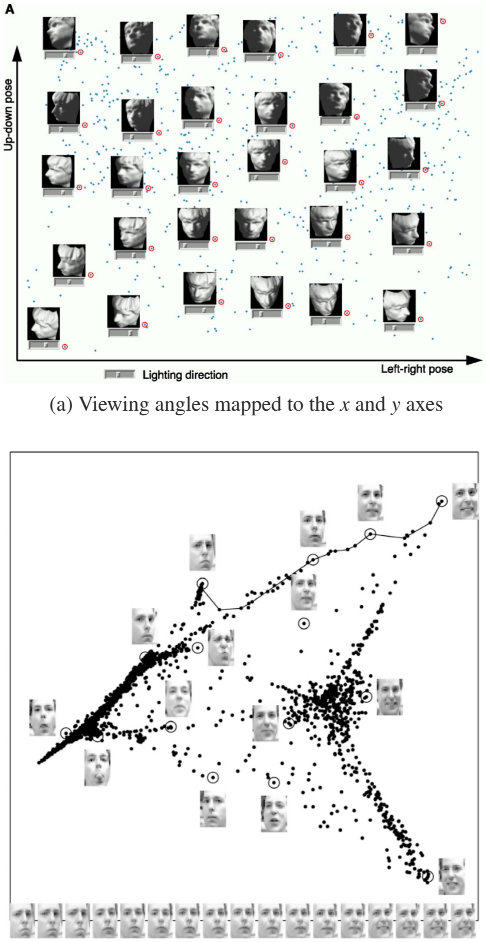
The new view in Fig. 11 shows her that she apparently enjoys visiting cities that have a rather high cost of living and have a percentage of population that has recently immigrated to the area. She explores the visualization, finding interesting insights such as Cleveland (on the top right) has low costs of living as it is high on her "dangerous" axis. In comparison, Springfield is much lower on the *y* axis, and has a similar cost of living. At this point, she continues to explore these data, fascinated by the data attributes that define the concepts of cities and areas that she has visited and that she did not consider prior to using this visualization. The application helped her realize that some of her concepts are grounded in the data, discovering attributes about the cities that help her describe subjective judgments, such as perceived safety.

5 Discussions

Thus far, we presented our novel interaction capabilities allowing users to interactively define an axis without much additional effort while exploring data in a scatterplot, along with several usage scenarios showing the effectiveness of the proposed interaction technique. In this section, we discuss the limitations and further improvements to overcome them.

5.1 Going Beyond Linear Models

Thus far, the proposed technique can characterize an axis as a weighted linear combination of the original features. Although this linear model enables us to fully maintain the interpretability, it suffers from the same limitation that other linear dimension reduction models have. What if the axis semantically meaningful to us is highly non-linear or curvi-linear? In machine learning and data mining, this issue has been actively studied in the context of non-linear dimension reduction [33] or manifold learning methods such as isometric feature mapping [42] and locally linear embedding [36]. In addition to their superior performances in various prediction tasks, many of these methods claim that the reduced dimensions generated by these non-linear methods may most likely correspond to some high-level, meaningful notions that can



(a) Viewing angles mapped to the *x* and *y* axes

(b) Facial expressions mapped to the *x* axis

Fig. 12. Non-linear dimension reduction examples using facial image data [42, 36].

be defined only in a curvi-linear coordinate system. For instance, in the case of facial image data as shown in Fig. 12, high-level characteristics such as a viewing angle and a facial expression (from frowning to smiling) have been mapped to the axis of the reduced-dimensional space [42, 36]. On the other hand, in a document data case, we may want to map something like the subjectivity or interestingness of an article to our axis [24].

However, in reality, because of many issues such as measurement noises and an insufficient number of data items, it is often too optimistic to expect a non-linear dimension reduction method to nicely map our high-level notions to an axis in a fully automated manner. Given this challenge, the proposed interaction method can open up a new possibility to interactively define such a non-linear high-level notion that we want to reveal. Typically, manifold learning methods define a curvi-linear axis forming the manifold surface approximately as a piece-wise linear model derived from the given high-dimensional data samples. If we extend the idea of user interaction proposed in this paper as a piece-wise linear model, then our interaction can actually allow users to define such a complicated manifold surface and its axis via an intuitive user interactions.

Users may perform an interaction of drawing an arbitrary, curved line passing through data items or groups of these items in a particular order, by which users could mean a progression of the above-mentioned characteristics such as document subjectivity. Then, our data-level interaction technique can be applied to form an axis as a piece-wise linear model (rather than a single linear representation) from the adjacent data items or groups on the path to approximately represent this curved line. In this manner, users can iteratively and interactively define even a highly non-linear axis through our data-level interaction technique.

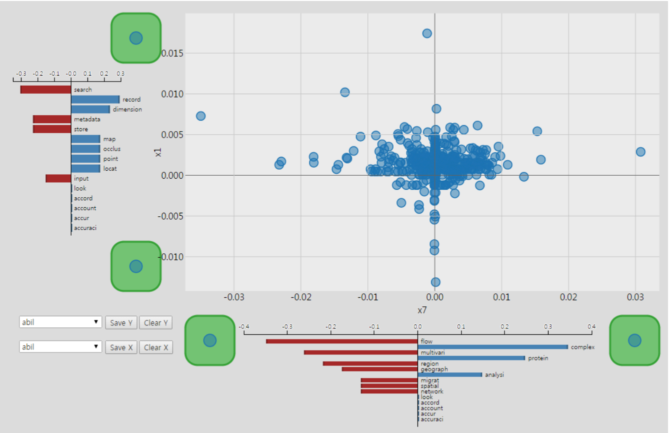


Fig. 13. An example of the proposed data-level interaction when applied to text data illustrating the challenge of applying the proposed technique to sparse data.

5.2 Handling Sparse Data

Another limitation of our interaction technique arises when our multi-dimensional data are sparse, meaning that there are few non-zero entries per dimension or data item. Such a sparsity issue is often found in significantly high-dimensional data such as images, text documents, and gene expression data. From the perspective of formality discussed in Section 3.4, since our axis can be quickly crystalized as a linear combination of (most of) the features even with a few data items, the proposed data-level interaction technique only needs a small amount of additional formality to define an axis.

However, this is no longer the case when each data item involves only a fraction of the features. In this case, one may have to assign considerably more data items to an axis than in the dense data case until reaching a desired level of specification of an axis. For example, Fig. 13 shows an example of text data when we assigned a data item to the high-end of an axis and another to the low-end. In this figure, one can see that many data items have been placed near the origin, indicating that the currently formed axis did not successfully show the variations of the other data items since the other items do not contain the keywords used for defining the axis.

One potential strategy to circumvent this sparsity issue against the proposed data-level interaction is to aggregate multiple dimensions into a single group. To this end, we can perform clustering on these dimensions on the basis of their co-occurrence patterns among data items, or alternatively, we can apply a dimension reduction method such as PCA, which can then provide a new definition of the reduced dimensions to start with as a linear combination of the original features. In this case, since our current interaction method also utilizes another linear model of these reduced dimensions, we can still maintain the full level of interpretability in terms of the original features.

5.3 Guiding Users towards Buried Information

This issue is more about a complementary approach to the proposed technique rather than about its limitation. To describe this issue, suppose that a user formed one axis at the moment, which mainly involves a particular set of features. For example, Fig. 6 shows that only the *y* axis has been specified, but the user may not know which other data items to choose to form another axis. In this case, given that the already defined *y* axis mainly involves features such as Wagon, AWD, RWD, Sports Car, and HP, the proposed system can recommend to the users data item pairs that can emphasize the other ignored features. In this manner, the system can help users build the two axes, each of which reveals complementary information about the data. Furthermore, considering the linear model that we currently adopt, this strategy can be viewed as choosing two axes that are orthogonal to each other, which is in common with the output PCA. Therefore, as an algorithm to recommend the data items for the next axis, we can exploit the established algorithm used in PCA such as the power iteration [25].

6 Conclusions

In this paper, we introduced InterAxis, a novel visual analytics technique to form and change an axis in a user-driven manner during the visual exploration of multidimensional data. By seamlessly incorporating data items to form the axis, the proposed technique expresses an axis as a weighted combination of the original features or attributes. Users can directly adjust these contributions/weights of attributes for each axis. In this manner, the proposed technique provides a direct way to specify axes through an interactive data exploration. To demonstrate the effectiveness of the proposed interaction techniques, we presented two usage scenarios by using real-world datasets such as car data and crime data. Finally, we discuss the potential limitations and the improvement strategies to overcome these limitations.

As our future work, we plan to improve the scalability of our system in terms of the computational cost of the underlying techniques as well as the visual clutter issues. We also plan to support the fine-tuning capabilities of the proposed system, such as assigning different weights to data items contained in a particular drop zone. The work presented in this paper advances our understanding of how to provide a user interaction with analytic models incorporated into visualizations.

7 Acknowledgments

This work was supported in part by DARPA XDATA grant FA8750-12-2-0309 and NSF grant CCF-0808863. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies. We also thank anonymous reviewers for their insightful comments and suggestions.

References

[1] C. Ahlberg. Spotfire: An information exploration environment. *SIGMOD Rec.*, 25(4):25–29, Dec. 1996.

[2] J. Alsakran, Y. Chen, Y. Zhao, J. Yang, and D. Luo. Streamit: Dynamic visualization and interactive exploration of text streams. In *Proc. the IEEE Pacific Visualization Symposium (PacificVis)*, pages 131–138, 2011.

[3] C. M. Bishop, M. Svensén, and C. K. Williams. GTM: The generative topographic mapping. *Neural computation*, 10(1):215–234, 1998.

[4] M. Bostock, V. Ogievetsky, and J. Heer. D3 data-driven documents. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 17(12):2301–2309, Dec. 2011.

[5] E. Brown, J. Liu, C. Brodley, and R. Chang. Dis-function: Learning distance functions interactively. In *Proc. the IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 83–92, 2012.

[6] H. Chernoff. The use of faces to represent points in k-dimensional space graphically. *Journal of the American Statistical Association*, 68(342):361–368, 1973.

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

[8] J. Choo, H. Lee, J. Kihm, and H. Park. iVisClassifier: An interactive visual analytics system for classification based on supervised dimension reduction. In *Proc. the IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 27–34, 2010.

[9] J. Choo, H. Lee, Z. Liu, J. Stasko, and H. Park. An interactive visual testbed system for dimension reduction and clustering of large-scale high-dimensional data. In *Proc. SPIE 8654, Visualization and Data Analysis (VDA)*, pages 1–15, feb 2013.

[10] J. Chuang, D. Ramage, C. Manning, and J. Heer. Interpretation and trust: Designing model-driven visualizations for text analysis. In *Proc. the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 443–452, 2012.

[11] J. Claessen and J. van Wijk. Flexible linked axes for multivariate data visualization. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 17(12):2310–2316, 2011.

[12] W. C. Cleveland and M. E. McGill. *Dynamic graphics for statistics*. CRC Press, Inc., 1988.

[13] W. S. Cleveland. *Visualizing Data*. Hobart Press, 1993.

[14] N. R. Draper, H. Smith, and E. Pownell. *Applied regression analysis*. Wiley, New York, 1966.

[15] N. Elmqvist, P. Dragicevic, and J.-D. Fekete. Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 14(6):1539–1148, 2008.

[16] A. Endert. Semantic interaction for visual analytics: Toward coupling cognition and computation. *IEEE Computer Graphics and Applications (CG&A)*, 34(4):8–15, 2014.

[17] A. Endert, P. Fiaux, and C. North. Semantic interaction for visual text analytics. In *Proc. the SIGCHI Conference on Human Factors in Computing Systems*, pages 473–482. ACM, 2012.

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

[19] Excel. *version 2013 (v15.0)*. Microsoft Corporation, Redmond, Washington, 2014.

[20] S. Few. *Now you see it: simple visualization techniques for quantitative analysis*. Analytics Press, 2009.

[21] I. K. Fodor. A survey of dimension reduction techniques, 2002.

[22] T. M. J. Fruchterman and E. M. Reingold. Graph drawing by force-directed placement. *Software: Practice and Experience*, 21(11):1129–1164, 1991.

[23] K. Fukunaga. *Introduction to Statistical Pattern Recognition, second edition*. Academic Press, Boston, 1990.

[24] M. Gleicher. Explainers: Expert explorations with crafted projections. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 19(12):2042–2051, 2013.

[25] G. H. Golub and C. F. Van Loan. *Matrix computations*. Johns Hopkins University Press, 2012.

[26] D. Jeong, C. Ziemkiewicz, B. Fisher, W. Ribarsky, and R. Chang. iPCA: An Interactive System for PCA-based Visual Analytics . *Computer Graphics Forum*, 28(3):767–774, 2009.

[27] I. T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, New York, 1986.

[28] D. Keim. Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 8(1):1 –8, jan/mar 2002.

[29] T. Kohonen. *Self-organizing maps*. Springer, 2001.

[30] B. Kondo and C. Collins. Dimpvis: Exploring time-varying information visualizations by direct manipulation. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 20(12):2003–2012, 2014.

[31] J. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29:1–27, 1964.

[32] J. LeBlanc, M. Ward, and N. Wittels. Exploring n-dimensional databases. In *Proc. the 1st conference on Visualization*, pages 230–237, 1990.

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

[34] MATLAB. *version 8.4.0 (R2014b)*. The MathWorks Inc., Natick, Massachusetts, 2014.

[35] K. A. Olsen, R. R. Korfhage, K. M. Sochats, M. B. Spring, and J. G. Williams. Visualization of a document collection: The vibe system. *Information Processing & Management*, 29(1):69–81, 1993.

[36] L. K. Saul and S. T. Roweis. Think globally, fit locally: Unsupervised learning of low dimensional manifolds. *Journal of Machine Learning Research (JMLR)*, 4:119–155, 2003.

[37] I. Shipman, FrankM. and C. Marshall. Formality considered harmful: Experiences, emerging themes, and directions on the use of formal representations in interactive systems. *Computer Supported Cooperative Work (CSCW)*, 8(4):333–352, 1999.

[38] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proc. IEEE Symposium on Visual Languages, 1996*, pages 336–343. IEEE, 1996.

[39] A. Skupin. A cartographic approach to visualizing conference abstracts. *IEEE Computer Graphics and Applications (CG&A)*, 22(1):50–58, 2002.

[40] D. F. Swayne, D. Temple Lang, A. Buja, and D. Cook. GGobi: evolving from XGobi into an extensible framework for interactive data visualization. *Computational Statistics & Data Analysis*, 43:423–444, 2003.

[41] Tableau. *version 8.1*. Tableau Software, Seattle, Washington, 2014.

[42] J. B. Tenenbaum, V. d. Silva, and J. C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290(5500):2319–2323, 2000.

[43] E. R. Tufte and P. Graves-Morris. *The visual display of quantitative information*, volume 2. Graphics press Cheshire, CT, 1983.

[44] C. Turkay, P. Filzmoser, and H. Hauser. Brushing dimensions - a dual visual analysis model for high-dimensional data. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 17(12):2591–2599, 2011.

[45] J. Utts. Seeing through statistics, 2005.

[46] L. van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[47] C. Ware. *Information visualization: perception for design*. Elsevier, 2012.

[48] J. A. Wise. The ecological approach to text visualization. *Journal of the American Society for Information Science*, 50(13):1224–1233, 1999.

[49] J. S. Yi, R. Melton, J. Stasko, and J. A. Jacko. Dust & magnet: multivariate information visualization using a magnet metaphor. *Information Visualization*, 4(4):239–256, 2005.

[50] J. Zhao, C. Collins, F. Chevalier, and R. Balakrishnan. Interactive exploration of implicit and explicit relations in faceted datasets. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 19(12):2080–2089, 2013.