The Generalized Sensitivity Scatterplot

Yu-Hsuan Chan, *Member, IEEE,* Carlos D. Correa, *Member, IEEE,* and Kwan-Liu Ma, *Fellow, IEEE*

Abstract—Scatterplots remain a powerful tool to visualize multi-dimensional data. However, accurately understanding the shape of multi-dimensional points from 2D projections remains challenging due to overlap. Consequently, there are a lot of variations on the scatterplot as a visual metaphor for this limitation. An important aspect often overlooked in scatterplots is the issue of sensitivity or local trend, which may help in identifying the type of relationship between two variables. However, it is not well-known how or what factors influence the perception of trends from 2D scatterplots. To shed light on this aspect, we conducted an experiment where we asked people to directly draw the perceived trends on a 2D scatterplot. We found that augmenting scatterplots with local sensitivity helps to fill the gaps in visual perception while retaining the simplicity and readability of a 2D scatterplot. We call this augmentation the generalized sensitivity scatterplot (GSS). In a GSS, sensitivity coefficients are visually depicted as flow-lines, which gives a sense of continuity and orientation of the data that provide cues about the way data points are scattered in a higher dimensional space. We introduce a series of glyphs and operations that facilitate the analysis of multi-dimensional data sets using GSS, and validate with a number of well-known data sets for both regression and classification tasks.

Index Terms—Sensitivity Analysis, Data Transformations, Model Fitting, Multidimensional Data Visualization

1 Introduction

Incorporating uncertainty and sensitivity analysis in visual analytics tools is essential to improve the decision-making process. On one hand, it provides the analysts a means to assign confidence levels to the insight gained through the analysis. On the other hand, it gives tool makers a methodology for measuring and comparing the robustness of data and visual transformations.

Sensitivity analysis (SA) refers to the analysis of small perturbations in the parameter space and their impact on the outputs. When we study pairwise correlations, sensitivity analysis tells us the rate of change of one variable Y with respect to another variable X. Scientists make use of such sensitivity studies to determine which input variables are more important or contribute more towards explaining the behavior of an output variable. Subsequently, these studies help reduce the parameter space to subspaces that are easier to analyze and visualize, and guide sampling strategies to obtain better samples.

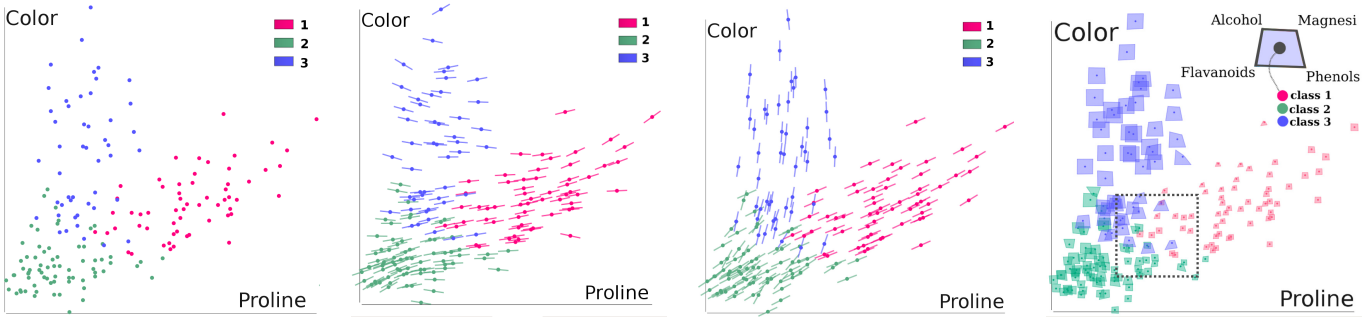
From a visualization standpoint, sensitivity analysis is augmented with two, often mutually exclusive, graphical representations: *sensitivity summaries*, which succinctly represent the outcome of a sensitivity study, and *sensitivity plots*, which usually provide detailed information, such as parallel coordinates or scatterplots with sensitivity information. Examples of sensitivity summaries include sensitivity matrices [14], where each cell in the matrix encodes the magnitude and direction of pairwise sensitivities between outputs and inputs; tornado diagrams [17] and box plots and their variants [33], [35]. Sensitivity plots, on the other hand, encode detailed information about derivatives and other statistical properties such as variances. In addition to visualization, interaction plays an important role for the analysis of sensitivity, as exemplified by interactive PCA [29] and ScatterDice [16]. However, relying on interactivity to explore data becomes impractical as the number of dimensions increases.

For this purpose, we propose to extend visualizations, such as scatterplots, with sensitivity information in a manner that does not preclude users from using the scatterplot in a familiar way, but that reveals aspects of the data that may not be evident without tedious interaction. In our previous work, we introduced the *flow-based scatterplot* (FBS) [10], which enhances scatterplots with sensitivity lines and streamlines, suggesting the appearance of flow to represent the relationship between two variables. FBS have a number of limitations, mainly the reliance on a 2D projection to compute the flow, which hides a lot of the complex interactions that may happen in the hidden extra dimensions, and the fact that the scatterplot onlys show the sensitivity with respect to a single variable at a time.

To alleviate these limitations, we have extended our contribution to what we call the *generalized sensitivity scatterplot* (GSS). To arrive at such representation, we first conducted a user-study to understand how people interpret trends from scatterplots of high-dimensional data. We show that, while certain trends may be evident from the distribution of points in a 2D projection, noise and complexity of the hidden dimensions make the task of interpreting a trend nearly impossible for complex data, unless the user resorts to thorough exploration of the extra dimensions.

We show that a generalized notion of sensitivity lines is a way to augment scatterplots in order to expose hidden interactions between variables. This generalization implies a subtle but fundamental modification to the FBS: flow-based scatterplots are by definition smooth and thus sensitivities are limited to the variables involved in a 2D projection, i.e., *differentiation occurs after projection*. In a GSS, sensitivities

- *Y.-H. Chan and K.-L. Ma are with the Department of Computer Science, University of California at Davis, Davis, CA, 95616. E-mail: chany@cs.ucdavis.edu and ma@cs.ucdavis.edu*
- *C. Correa is with Lawrence Livermore National Laboratory, Livermore, CA. E-mail: cdcorrea@gmail.com*



(a) Scatterplot (b) Flow-based scatterplot (c) GSS (d) Sensitivity Star Glyphs

Fig. 1. Visualization of two variables of the Wine data set [47] where points are color-coded by class. (a) Traditional scatterplots suffer from overlap and different classes may appear to mix in arbitrary ways. (b) a FBS [10] reveals a positively correlated trend, but mis-represents the trend of the points in blue. (c) a GSS on a 3D subspace represents better the trends of points belonging to different classes; class 3 is distinguishable from others. (d) A star glyph plot summarizes the GSS across multiple 3D subspaces, where all of the three classes stand out distinctly, as evidenced by the shape and size of the glyphs.

involve the full parameter space or a selected subspace, not necessarily the same as the projection, i.e., *projection occurs after differentiation*. We show that this operation implies a number of relationships between the visualization of partial derivatives in a 2D subspace and what can be inferred about the smoothness of the data in higher dimensions.

In addition, we found that we can interpret sensitivity in a more abstract way, and we do not need to limit ourselves to a single variable to represent the sensitivity of data points. Instead, we can consider multiple sensitivities at once and represent these using 2D glyphs. We call these the *sensitivity fan* and the *sensitivity star glyph*, which encode the direction and magnitude of sensitivity for multiple variables, respectively. We show in a number of examples that these glyphs help visually classify data in ways that are not possible unless we consider the full parameter space. To illustrate the contributions of our paper, we now describe an example of how the GSS and sensitivity glyphs are used.

1.1 An Illustrative Example

Let us consider the wine data set that comprises 13 variables of 178 observations of the chemical composition of wines growing in Italy and the relationship to color intensity and hue, as described in [47]. They are classified into three categories shown in red, green and blue. A scatterplot of variables *proline* and *color* is shown in Fig. 1(a), with color encoding the *class* of wine. One way to visualize sensitivity is via flow-based scatterplots (FBS), as shown in Fig. 1(b), where line segments indicate a sense of the local trend of the data. However, vertical regression, used to extract sensitivity parameters, emphasizes functional relationships, which, while correctly showing the relationship between *color* and *proline* for the points in red, forces the points in blue to be interpreted as having the variable *color* decrease with the variable *proline*. Our GSS provides a more general view of sensitivity. In Fig. 1(c) we depict two contributions: (1) sensitivity lines are computed using orthogonal regression, which allows us to discover non-functional relationships (or functional relationships of the X variable with respect to Y), as seen for the points in blue; and (2) sensitivity lines are projected from a higher dimensional space, allowing us to see two overlapping trends (one formed by blue points and another formed by green and red points), exposing a relationship in a third dimension otherwise hidden in the traditional and FB scatterplots.

While sensitivity lines show quantitative properties of the sensitivity, we can think of sensitivity in a more abstract manner. One mechanism is to consider the magnitude of sensitivity along different subspaces as additional dimensions of the data. Then, one can replace points in the scatterplot with glyphs, such as a sensitivity star, described in more detail in Sec. 5.2.3. Fig. 1(d) shows a GSS of *proline* vs. *color* with sensitivity star glyphs, in which each point is augmented with a quadrilateral where each vertex is at a distance from the data point proportional to the magnitude of the sensitivity in a different subspace. The four subspaces are formed by the variables *proline*, *color*, and one of the extra dimensions: concentration of *magnesium*, *alcohol*, *flavanoids* and *phenols*. This augmentation shows that representing each point by a glyph provides visual separation of the high dimensional points in a simpler 2D plot. Notice how points in the overlap of the three classes have very distinct shapes and sizes that are similar within each class, but quite different among classes, which is an ideal property for effective classification.

2 Related Work

Multivariate analysis. Multivariate analysis is at the core of visual analytics. Approaches can be categorized as data-centered approaches, such as regression [15], generalized additive models [24] and response surface analysis [6], or visual-centered approaches. Since data is becoming large and complex, data-driven approaches often employ simplification techniques, which either reduce the number of observations, such as binning, sampling [42] or clustering [5], or reduce the number of dimensions in the data, such as projections [37] and multi-dimensional scaling. Visual-centered approaches follow a different strategy, where correlations and trends emerge as salient structures in the human visual system. These approaches are often coupled with interactive manipulation, as

shown by Jeong et al., who incorporate interactivity into principal component analysis [29]. Yang et al. integrate analysis tools with the visual exploration of multivariate data [49]. In this paper, we present a combination of analysis and visualization tools that exploit sensitivity analysis for the effective exploration and navigation of multi-dimensional data.

Sensitivity analysis. There are numerous approaches to studying sensitivity, including *local analysis* [8], [21], where the sensitivity parameters are found by simply taking the derivatives of the output with respect to the input; *statistical* methods, such as those based on variance, which provide an estimate of the sensitivity in terms of the probability distribution of the inputs [9], [28], [1]; or *sampling-based* methods, when it is not feasible to sample the entire parameter space, including methods such as Latin hypercube sampling [27] and Montecarlo simulations [40], [26].

For surveys on sensitivity analysis methods, including their application to multivariate analysis, refer to Frey and Patil [19] and Tanaka [41]. Sensitivity analysis can also be described as a general recipe for analyzing specific data tools, such as variance analysis [9], clustering [13], [11], principal component analysis [39], [48] and uncertainty analysis [31].

Sensitivity analysis in visualization. Recently, it has become important to visualize sensitivity parameters along with the data. Barlowe et al. [3] proposed the use of histograms and scatterplot matrices to visualize the partial derivatives of dependent variables to reveal the type and strength of correlations between the output and the inputs of a process. Correa et al. [14] used sensitivity analysis to propagate the uncertainty in a series of data transformations and proposed a number of extensions to show this uncertainty in 2D scatterplots. We explored further this notion with flow-based scatterplots [10]. Bachthaler et al. [2] presented the continuous scatterplot, which generates a continuous density function for a scatterplot and alleviates the issues with missing data. Heinrich et al. [25] extended this to parallel coordinates, while Feng et al. [18] incorporated uncertainty analysis to provide density-based views of multivariate data.

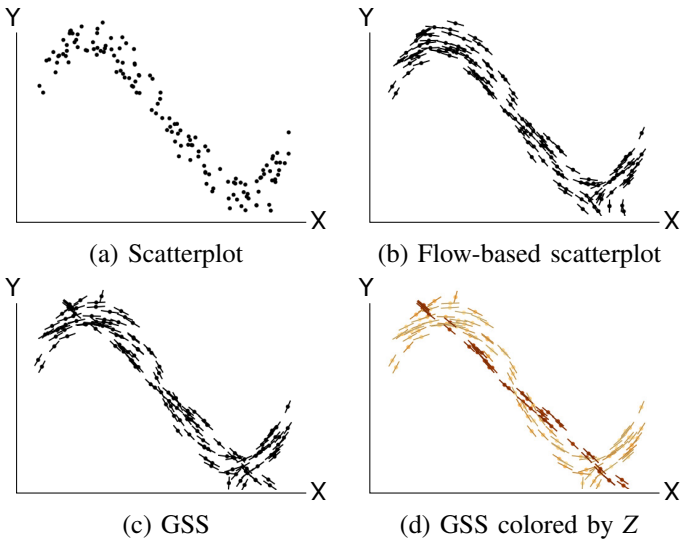
FBS are constructed in a similar fashion, by estimating a density function that explains the 2D plot. Rather than a global density, we implicitly fit a density function locally to estimate the derivative of a function at each point. Guo et al. [22] extend the visualization of sensitivity analysis to local views, where the analyst is able to compare local sensitivity coefficients in a myriad of tools. Berger et al. [4] estimate local neighborhoods to represent sensitivity with respect to multiple variables as overlapping areas in a scatterplot. The shape and size of these areas give an idea of how a function changes locally. In this paper we also address the issue of visualizing multiple sensitivities simultaneously. Using sensitivity coefficients as data dimensions allows us to use existing glyphs to create effective scatterplots. A related representation is the spiderplot by Eschenbach et al. [17], who used it to show the relative change in the outcome for a unit change in multiple independent variables. Their work inspired the sensitivity fan and star glyphs proposed in this paper.

On augmenting and interacting with scatterplots. Scatterplots are intuitive to understand when studying the relationship between two variables. However, projected points may result in clutter and overlap for large and high dimensional data sets. To deal with the loss of information that comes from projection, a number of techniques are proposed. Keim et al. [30] proposed generalized scatterplots that let users balance between the amount of overlap and distortion of the data points. Other augmentations have been proposed by Collins et al. [12], who enhance the spatial layout of plots with clustering information, and Shneiderman et al. [38], who link multiple substrate plots to superimpose cross-substrate relationships. Other techniques rely on interaction and navigation. One mechanism is to link projections in a scatterplot matrix in order to enumerate all possible combinations of projections of variables, but an effective way to navigate these spaces remains a challenge. Scatter dice [16] is an alternative that exploits interactive capabilities to navigate a large scatter matrix and help visual analytics. Tools also often provide selection and brushing techniques to aid in the exploration of high-dimensional spaces. Aside from axis-aligned selections, ubiquitous in interactive tools, there has been some work in more meaningful brushes, such as structure-based brushes [20] and n-dimensional brushes [32]. In our work, we show that sensitivity information can be considered as additional important dimensions of the data, and so can be used to select related data in a more meaningful way. We aim to highlight the importance of local sensitivity coefficients in aiding visual analytics, and so we show that existing techniques, such as selection, brushing and navigation, are enhanced in meaningful ways. For this reason, we do not survey the extensive literature in interactive techniques for scatterplots.

3 Sensitivity and Scatterplots

Common scatterplots, i.e., depicting only 2D points, only implicitly encode the sensitivity of one variable with respect to the other, as depicted in Fig. 2(a). To quantify the sensitivity of one variable with respect to another, we recur to regression. Typically, linear regression, obtained via *total least squares*, provides a single value of linear sensitivity or correlation. However, when the two variables are not related via a linear function, such as shown in Fig. 2(a), a single regression line hides the true nature of the interaction between the two variables. An alternative is to compute local regression, e.g., using *weighted least squares* or *locally weighted polynomial regression* [34], which makes it possible to quantify complex trends in the data, which, although locally linear, may exhibit nonlinearities when viewed as a whole.

A visual representation of sensitivity is what we call a sensitivity-enhanced scatterplot. Trend lines are present in practically all scatterplot graphing tools. Recently, flow-based scatterplots (FBS) [10] extend this idea to local regression, encoding a local regression line for each data point in the 2D plane. In this paper, we present a generalization, called generalized sensitivity scatterplots (GSS), which decouples the projection step during the construction of the scatterplot and the regression step that computes the sensitivities. This is important in analyzing multi-dimensional data when we aim to find relationships that cannot be explained by two variables.



(a) Scatterplot (b) Flow-based scatterplot (c) GSS (d) GSS colored by Z

Fig. 2. Scatterplots of a synthetic function. (b) the sensitivity shows a sinusoidal function. (c) the trend lines show both the sinusoidal and linear components only visible in the hidden Z dimension, evident in (d).

An example is shown in Fig. 2(c-d), which depicts a projection of a 3D function that combines both a linear and a sinusoidal relationship, as explained below.

First, we introduce the basics of sensitivity estimation in high dimensional spaces.

3.1 Definition

Let $\mathbf{x} \in \mathbb{R}^N$ be a point in an $N-$dimensional space. The sensitivity of $\mathbf{x}$ with respect to a variable x_i is simply the partial derivative

$$\mathbf{y} = \frac{\partial \mathbf{x}'}{\partial x_i} \tag{1}$$

For a FBS [10], $\mathbf{y}$ is a 2D vector that results from computing the sensitivity of a projected point $\mathbf{x}' = (x,y)^\top = \Pi_S(\mathbf{x})$, where $\Pi_S : \mathbb{R}^N \mapsto \mathbb{R}^2$ is a projective transformation. In other words, FBS applies the differentiation step *after* projection.

In general, however, one must decouple the projection and subspace selection transformations from the differentiation operation for the sensitivity analysis to be effective.

Let us consider, without loss of generality, $\mathbf{y} = (u,v)$ a 2D sensitivity tuple associated with a point $\mathbf{x}$, computed as

$$(u(\mathbf{x}), v(\mathbf{x}))^\top = \Pi_D \left(\frac{\partial \Pi \mathbf{x}}{\partial x_i} \right)^\top \tag{2}$$

for a transformation $\Pi_D : \mathbb{R}^M \mapsto \mathbb{R}^2$, called the *projection* and a general transformation $\Pi : \mathbb{R}^N \mapsto \mathbb{R}^M$, $M \leq N$, called the *subspace selection*, which can be the identity transformation or another projection.

From Eq. 2, we see that, when $\Pi = \Pi_S$ and Π_D is the identity transformation, the GSS becomes a flow-based scatterplot.

3.1.1 Example

To understand the importance of this generalization, let us consider an example of a set of 3D points (x,y,z), where $y = f(x)(1-z) + g(x)z$. Fig. 2(a) depicts the projection of this set of points in the XY plane for $f(x) = sin(\alpha x)$ a sinusoidal and $g(x) = ax + b$ a linear function. From the projection, it is difficult to see that this is indeed the shape of the 3D function. To visualize this relationship, we attempt to obtain two different sensitivity-based scatterplots, one using the flow-based scatterplot and the other one using our generalization.

Let us consider the case of differentiation *after* projection in the XY plane (or $z = 0$). In this case, the newly projected points (x', y') become $(x, f(x))$. Then, the vector lines are formed by the derivatives $(\frac{\partial x}{\partial x}, \frac{\partial y}{\partial x}) = (1, \frac{\partial f}{\partial x})$. This approach hides the interaction of x and z to form function y. As a result, as shown in Fig. 2(b), the vector lines only show the sinusoidal aspect of the function, i.e., $f(x)$.

Now let us consider the case of differentiating the function *before* projection. The derivatives of a point are $(\frac{\partial x}{\partial x}, \frac{\partial y}{\partial x}, \frac{\partial z}{\partial x}) = (1, \frac{\partial f}{\partial x}(1-z) + \frac{\partial g}{\partial x}z, \frac{\partial z}{\partial x})$ and their projection in the 2D plane becomes: $u = 1$ and $v = \frac{\partial f}{\partial x}(1-z) + \frac{\partial g}{\partial x}z$.

Thus, this vector line is now able to represent the interaction with the hidden coordinate z, as depicted in Fig. 2(c-d).

To understand better the importance of exposing hidden factors when computing sensitivities, we conducted a user study where we let subjects draw trends of synthetic data from looking at a 2D scatterplot of a sampled 3D function. We used the results of this study to identify the effect of noise and complexity of the function and justify the need for explicit depictions of sensitivity.

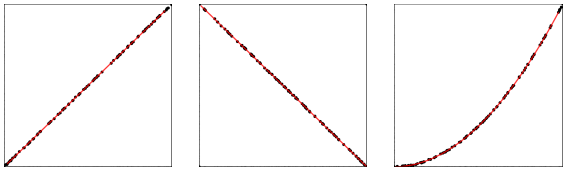
4 How People Interpret Trend: An Empirical Study

We conducted a user study to help us understand how people interpret trends from scatterplots. This focus allowed us to conduct controlled experiments and gather an unprecedented data set that will be useful for future research on trends and scatterplots.

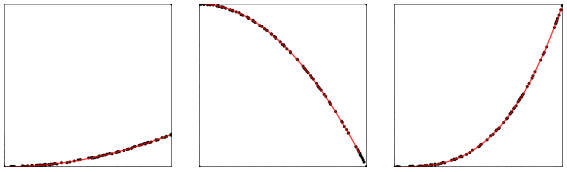
We narrowed our evaluation to the understanding of trends of functions in three dimensions given a single two-dimensional projection. A 2D scatterplot of this 3D function gets more difficult to understand when the function is perturbed by noise, when the function along the hidden dimension increases the ambiguity of the function, or when the functions in the 2D plot are complex.

We generated a number of 3D functions formed by two functional relationships interpolated smoothly along a hidden dimension and sampled the functions at discrete positions. We asked users to look at the 2D scatterplot of these functions and identify up to two trends that best describe the underlying function. If we gather enough results for each function, we can visualize and quantify the amount of agreement the different users have in terms of the shape and location of those trends. We hypothesize that the level of agreement decreases as the level of noise increases and is less for a smooth interpolation instead of a sharp separation of the function values.

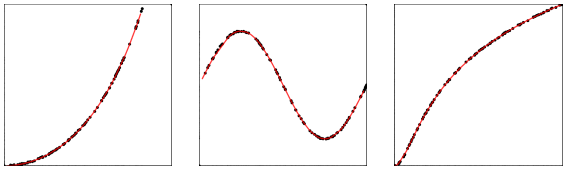
In the cases where it is difficult to infer a trend from a scatterplot alone, the generalized sensitivity plot will be more informative and will alleviate the perceptual and cognitive ambiguities that arise from the projection.



(a) LIN1: x (b) MLIN: $1-x$ (c) QUAD: x^2

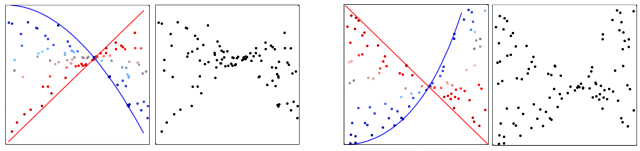


(d) QUAD2: $0.3x^2$ (e) MQUAD: $1-x^2$ (f) CUB: x^3

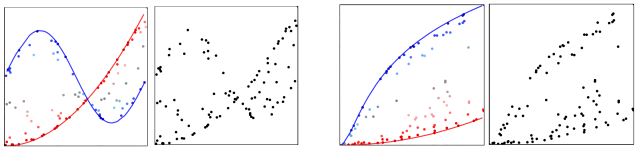


(g) EXP: $e^{-x^2}-1$ (h) SIN: $\sin 2\pi x$ (i) LOG: $\frac{\log(ax^2+1)}{\log(a+1)}$

Fig. 3. Nine of our ten patterns in the study. Top: LIN,MLIN,QUAD. Middle: QUAD2, MQUAD, CUB. Bottom: EXP, SIN, LOG. Not seen here is LIN2: $1.5x$.



(a) LIN1 & MQUAD (b) MLIN & EXP



(c) QUAD & SIN (d) QUAD2 & LOG

Fig. 4. 3D functions interpolated from seed functions (red and blue curves) and an interpolant. (a-b) use linear interpolant and (c-d) are from sigmoid interpolant. In the image on the left, points are colored by the interpolation weight w. The monochromatic image on the right is the final image shown to the participants.

4.1 Data Preparation

We create a set of synthetic 3D functions as follows:

$$Y(x,z) = w(z)Y_1(x) + (1-w(z))Y_2(x) + N_\alpha(x) \qquad (3)$$

where Y_1 and Y_2 are two 1D functions we call *seeds*, w is a monotonically increasing *interpolating* function and N_α is a noise function of amplitude α. In this experiment, we chose a uniform noise function, which produces real values in the range $[-\alpha,\alpha]$. Fig. 3 summarizes the functions we used as seeds.

We have generated 45 combinations of these patterns, each with two interpolating functions w: Linear (LIN) and Sigmoid (SIG). For a number of tasks, we have also generated 10 functions with a single seed, i.e., for a unit step function as an interpolation function. In total, this amounts to 100 different 3D functions. Examples of these combinations are shown in Fig. 4(a-e). Note that users were only presented with uncolored 2D scatterplots, where the seed functions are not necessarily obvious. Besides the seed functions and the interpolation functions, we also introduce three different levels of noise $\alpha \in \{0.01, 0.13, 0.25\}$.

4.2 Design of the User Study

We want to learn how users interpret local trends from the scatterplot. In order to do that, we expected that we would need at least ten experimental runs per function. These runs, coming from different users, can be overlaid on the scatterplot to generate a *confidence plot* for how easy it is to infer trend from data. We also limited each user to 20 tasks to avoid learning effects and minimize performance degradation for any given user. We estimated that we needed at least 150 participants for the study, so we developed an online tool that we made available to volunteers.

After sending invitations to participate, we recruited 223 people. From the total submitted tasks, we identified a set of 6140 valid results after removing some outliers, which resulted in 10.77 valid results per plot. A total of 134 males and 89 females participated: 132 of them had graduate or similar level of education, 150 participants were in the 20-30 age group, and 167 people had an Engineering or Science major.

Subjects filled a questionnaire about their background first, and were exposed to 20 randomly selected tasks from the 100 sets of different trend plots as shown in Fig. 5. The noise level and the interpolation function for each task were randomly selected while guaranteeing that the total number of tasks for each difficulty level was similar in order to avoid cases where tasks are all simple (low noise level) or all difficult (high noise level). In each task, subjects were asked to draw freely a curve that best describes each trend they perceived in the plot, and were allowed to unlimited re-dos if they were not satisfied. For each task, we collected:

- The total time to finish the task and to to draw curves.
- The number of re-do for each curve and the curves drawn.

All the subjects were asked to make an intuitive decision on the trends they perceived in the plot of mixed pattern. Before the study, the subjects knew nothing about the generalized sensitivity nor the seed functions we used to generate mixed trends. They did not know the answers of the trends mixed in the plots until the end of the study.

To alleviate any technical difficulty, the subjects were given a practice session where they were shown the same drawing interface in Fig. 5 and it was not timed. The subjects were allowed to use their preferred input device such as a mouse, a touch screen, or a pen stylus. We asked the user to explicitly identify their input method. Among the 223 subjects, 180 used their mouse to draw, while only 27 used a more intuitive way to draw such as a stylus pen or their finger on the display.

We show the results of our study in two forms: (1) Confidence plots, which summarize all the drawings from participants for each function, and (2) The average error of

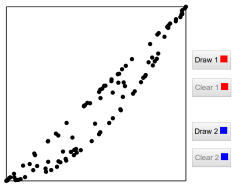


Fig. 5. A snapshot of the online user study.

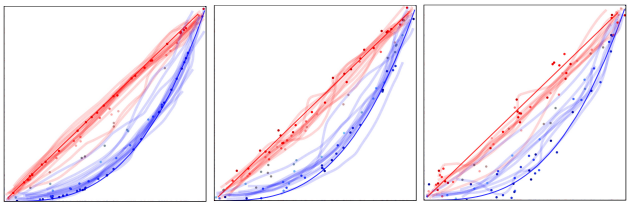


Fig. 6. Confidence plots as noise increases.

the perceived trends compared to the seed functions and the local trend given by their local sensitivity.

4.3 Confidence Plots

The confidence plot for a function is constructed by overlaying the responses for the different subjects that encountered that function using semi-transparency. We pre-processed the raw sketches by smoothing the drawing (with the same parameter), color coding each sketching curve to the corresponding seed curve, and removing outliers. Since we made no assumption of the trend lines that subjects might draw, and we encouraged subjects to draw the most intuitive trends they saw from the plot, a few of the results included loops and non-functional relationships, which we excluded as outliers. Some example of the confidence plots are shown in Fig 8. The complete set of the confidence plots are available at *http://vidi.cs.ucdavis.edu/projects/RegressionStudy*. We identified noticeable differences in the dimensions considered in our study:

- **Noise**: the larger the noise, the lower the consensus. For a set of three functions with the same seeds and interpolation but different noise levels, it is clear that participants agree less. An example is shown in Fig. 6.
- **Interpolation**: functions interpolated using a sigmoid interpolant have more consensus than those using a linear interpolant. This illustrates one of the difficulties in inferring trends from 2D projections. While it may be easy when points are clearly separated in the hidden dimension, it becomes increasingly difficult to infer trends for smoothly changing functions. An example is shown in the third (sigmoid) and fourth (linear) images in Fig. 7.
- **Single or Mixed Seed functions**: also shown in Fig. 7 as a special interpolant with a unit step function.

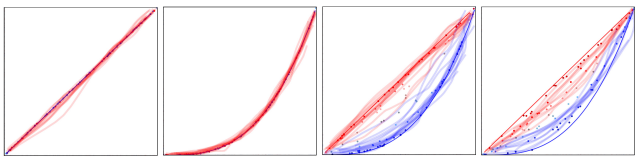


Fig. 7. Confidence plots in terms of interpolation along the hidden dimension.

Rank	function	Mean	Min (inter,noise)	Max (inter,noise)
1	LIN	0.0904	0.0252 (SIG,M)	0.1805 (LIN,S)
2	LOG	0.0904	0.0182 (one,S)	0.2144 (LIN,M)
3	EXP	0.0935	0.0295 (LIN,M)	0.2429 (LIN,M)
4	QUAD	0.0936	0.0187 (one,S)	0.2398 (LIN,M)
5	MLIN	0.0993	0.0178 (one,S)	0.2453 (LIN,M)
6	SIN	0.1038	0.0198 (one,S)	0.1969 (LIN,S)
7	LIN1	0.1043	0.0304 (SIG,L)	0.1909 (LIN,L)
8	CUB	0.1145	0.0197 (one,S)	0.2777 (LIN,S)
9	MQUAD	0.1204	0.0242 (one,L)	0.2182 (LIN,M)
10	QUAD2	0.1617	0.0312 (one,S)	0.3537 (LIN,S)

TABLE 1
Mean, Min and Max Error of the distance index E.

4.4 Performance

In addition to qualitative evaluation, we also study the quantitative performance of the user perception of trends versus the ground-truth trend given by either the seed functions or the local sensitivity illustrated by generalized sensitivity plots.

4.4.1 Perceived Trend vs. Seed Functions

We estimate the average error for each function Y as the Euclidean distance of a point in the curve to the closest point in the corresponding seed function. We denote the error as

$$E(Y,C,\alpha,w) = \sum_{x \in C} \min d(C(t),Y(t)) \qquad (4)$$

for a user drawn curve C parameterized by t and the curve Y defined by a seed function. The total error for a function is simply the average among the available curves: $E(Y,\alpha,w) = \frac{1}{N}\sum_i E(Y,C_i,\alpha,w)$ for a set of N curves C_i.

To summarize the error for each seed, we also averaged the error among different values α and interpolations functions w. The resulting averages, along with the minimum and maximum noise and interpolation configuration, is shown in Table.1 and depicted in Fig. 8. Notice that for seven out of ten functions (LOG, QUAD, MLIN, SIN, CUB, MQUAD, and MQUAD2), the minimum error took place in the respective plot with a single seed. For the other three functions (LIN, EXP, and LIN1), their minimal error took place at plots that are either interpolated by SIGMOID function (LIN and LIN1), or at plots with a smaller noise level (EXP with Medium noise). Larger errors, as observed before, appeared when using the linear interpolant.

A 2-way ANOVA on the effects of noise and interpolation showed a significant effect of noise F = 6.94 (p=0.0011) and a significant effect of the interpolant, F = 180.31 (p <0.0001), validating our hypotheses with much higher confidence.

4.4.2 Perceived Trends v.s. Local Fit

Because users may interpret local trend in different ways, we measured the degree to which the curves drawn by the subjects appear to locally fit the data. We estimate a similar error for each function Y as the error between the a point in the drawn curve and a sample point. We denote the error as $E_F(Y,C,\alpha,w)$, and computed similar averages and bounds. The confidence plots of their min and max error are summarized in table 2 and shown in Fig. 9. Finally, to see how well the curve agrees with the local fit, we treat the seed

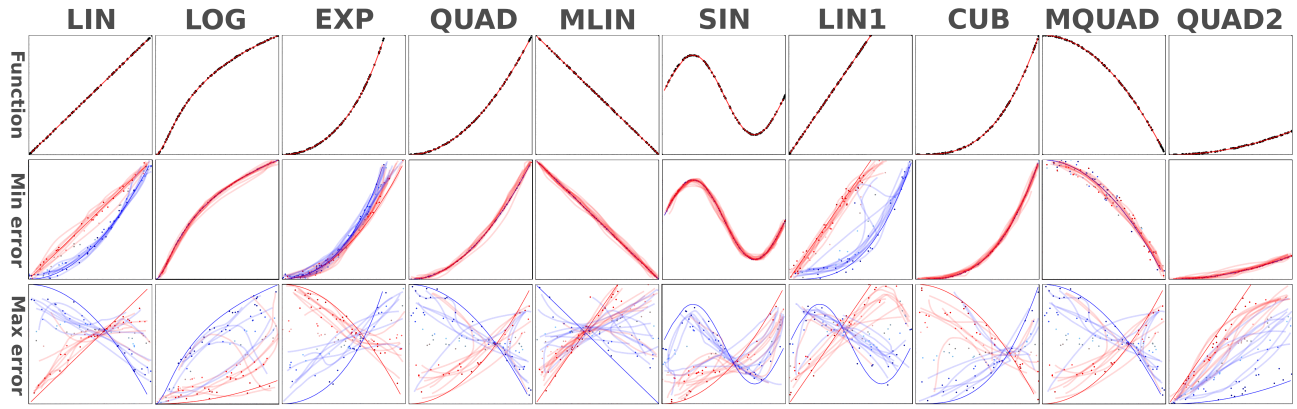


Fig. 8. The confidence plots of the minimal and maximum E.

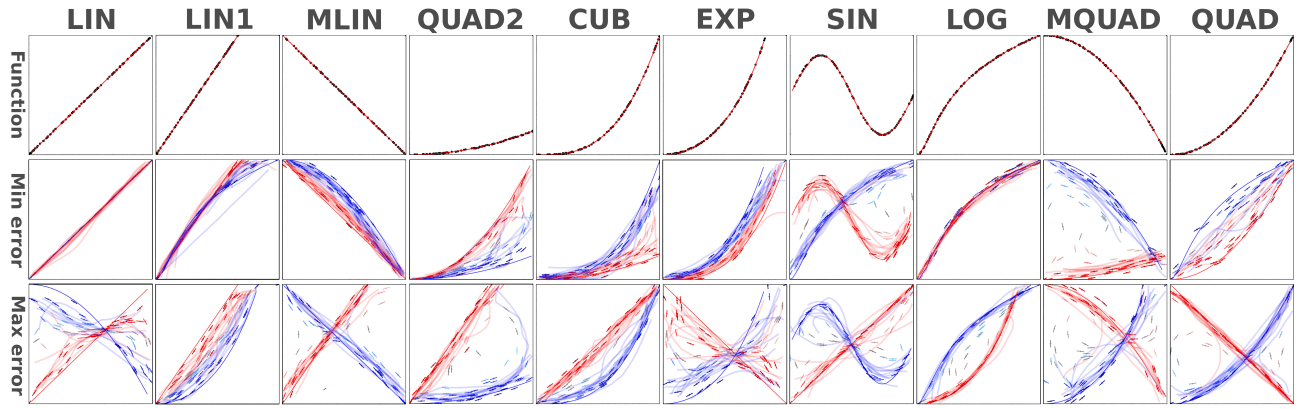


Fig. 9. The confidence plots of the minimal and maximum E_F.

Rank	function	Mean	Min (inter,noise)	Max (inter,noise)
1	LIN	0.0774	0.0173 (one,S)	0.1981 (LIN,S)
2	LIN1	0.0908	0.0253 (LIN,S)	0.2328 (LIN,S)
3	MLIN	0.1063	0.0341 (LIN,M)	0.2037 (SIG,M)
4	QUAD2	0.1068	0.0565 (LIN,S)	0.1884 (SIG,M)
5	CUB	0.1103	0.0464 (LIN,M)	0.1861 (SIG,M)
6	EXP	0.1107	0.0609 (LIN,M)	0.1764 (LIN,L)
7	SIN	0.1162	0.0417 (SIG,M)	0.2375 (SIG,S)
8	LOG	0.1233	0.0216 (one,M)	0.1956 (SIG,S)
9	MQUAD	0.1263	0.0615 (SIG,M)	0.2109 (SIG,L)
10	QUAD	0.1290	0.0353 (LIN,L)	0.2307 (SIG,S)

TABLE 2
Mean, Min and Max Error of the distance index E_F.

Rank	function	Mean	E	E_F	E_S
1	LIN	0.0847	0.0905	0.0774	0.0863
2	LIN1	0.0968	0.1043	0.0908	0.0954
3	EXP	0.1062	0.0935	0.1107	0.1144
4	MLIN	0.1076	0.0993	0.1063	0.1173
5	CUB	0.1156	0.1145	0.1103	0.1220
6	SIN	0.1164	0.1038	0.1162	0.1293
7	LOG	0.1175	0.0905	0.1233	0.1388
8	QUAD	0.1200	0.0937	0.1290	0.1373
9	QUAD2	0.1257	0.1617	0.1068	0.1086
10	MQUAD	0.1289	0.1205	0.1263	0.1398

TABLE 3
Means of the three distance indexes: E, E_F, and E_S.

function as a baseline curve and compute the error as the Euclidean distance between points in the seed function and the closest sensitivity line, denotes as E_S. Table 3 summarizes the three types of errors: E between a seed function and drawn curves, E_F between the sensitivity lines at the sampled points and drawn curves and E_S between the seed function and the sensitivity lines. For any given function when E_S is closer to E than E_F, users interpret trends more in a global manner, as a fitting of a single perceived function. Otherwise, users interpret trends more locally and in an adaptive manner, as a function of the hidden dimension.

4.5 Discussion

We have set out to qualitatively and quantitatively measure how much agreement there is between different users about what constitutes a local trend. While this is a seemingly simple task, there is little consensus about what constitutes a trend when data is corrupted by noise, or obvious trends are obscured by the projection when data varies along an unknown dimension. In those cases, it is imperative to augment the scatterplot to explicitly depict the local trend and alleviate any ambiguities perceived by users. Generalized sensitivity scatterplots provide such augmentation, as we will describe in the next section.

5 Generalized Sensitivity Scatterplots (GSS)

A GSS is a graphical representation of N-dimensional data that represents a data point $\mathbf{x} \in \mathbb{R}^N$ via a tuple $(x, y, u(\mathbf{x}), v(\mathbf{x}))$

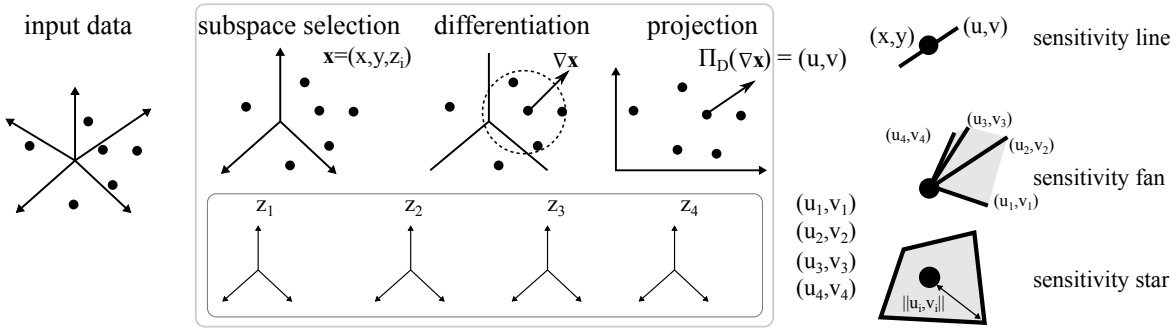


Fig. 10. Process of computing a GSS. We start by computing sensitivities from the input data on a selected subspace. After approximating the derivative for each point in that subspace, we project to a 2D space. The resulting sensitivities are visually encoded as sensitivity lines in the general case. For multiple sensitivities, we can encode them simultaneously using the fan or the star glyph.

where (u,v) is computed using Eq. 2. The process of computing a GSS from raw data to a visual representation is depicted in Fig. 10, and described in the following sections. The first stage computes the sensitivity coefficients (u,v) using the method outline in Eq. 2, which comprises three transformations: subspace selection, differentiation with respect to a variable of interest, and projection. Because we start from discretely sampled data and the underlying function is not always known, we start by estimating the partial derivatives numerically.

5.1 Estimating Partial Derivatives

In this paper we follow the approach in our previous work [10] and approximate the partial derivatives using a variational approach, where we use the slope of the locally fitted linear regression around a given point, by taking in consideration the Taylor expansion of a given variable y with respect to another variable x. For a point (x_0, y_0),

$$y_i - y_0 \approx \frac{\partial y}{\partial x}|_{(x_0,y_0)}(x_i - x_0) \tag{5}$$

For a set of k points (x_i, y_i), it can be posed as a linear problem:

$$W \begin{bmatrix} x_1 - x_0 \\ \vdots \\ x_k - x_0 \end{bmatrix} \beta = W \begin{bmatrix} y_1 - y_0 \\ \vdots \\ y_k - y_0 \end{bmatrix} \tag{6}$$

$$WX\beta = WY \tag{7}$$

where $W = \mathrm{diag}\{w_i\}$ is a diagonal matrix of weights, representing the importance given to any point i, and β (variable to solve), is the partial derivative $\frac{\partial x}{\partial y}$. In general, these weights are inversely proportional to the distance to the target point (x_0, y_0). We use a Gaussian weighting scheme where $d(\mathbf{x}_0, \mathbf{x}_i)$ is the Euclidean distance between two points in $\mathbb{R}^M$:

$$w_i = e^{-d(\mathbf{x}_0,\mathbf{x}_i)^2} \tag{8}$$

This problem can be solved using linear least squares [45], by solving the linear system

$$X^\top W^2 X\beta = X^\top W^2 Y \tag{9}$$

This approach, however, fits a hyperplane using vertical offsets along the y dimension, and fails to represent partial derivatives perpendicular to the x dimension, as evidenced by our previous work [10]. In this work, we adopt the notion of orthogonal least squares [46], where the quantity to be minimized is the orthogonal distance to the hyperplane defined by the partial derivative by solving the quadratic problem:

$$\beta^2 + C\beta - 1 = 0$$

$$C = \frac{S_{yy} - S_{xx} + (S_x^2 - S_y^2)/S_w}{S_x S_y / S_w - S_{xy}}$$

$$S_x = \sum_i w_i(x_i - x_0) \quad S_y = \sum_i w_i(y_i - y_0)$$

$$S_{xx} = \sum_i w_i(x_i - x_0)^2 \quad S_{yy} = \sum_i w_i(y_i - y_0)^2$$

$$S_w = \sum_i w_i \quad S_{xy} = \sum_i w_i(x_i - x_0)(y_i - y_0)$$

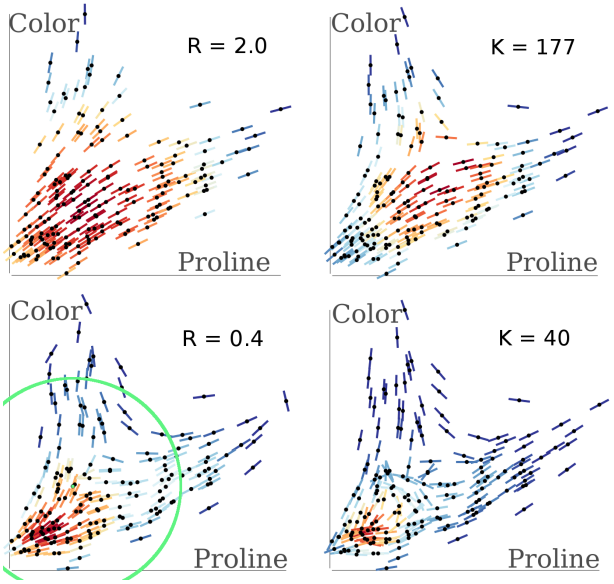
Fig. 1(b-c) show the difference between vertical and orthogonal regression. While local trends using vertical regression emphasize functional relationships (and thus tend to align with the X axis), orthogonal regression is more effective at capturing non-functional relationships, including trends that are not aligned with the X axis, such as the trend of class 3 points in blue.

5.1.1 Setting the Neighborhood Size

As seen above, we can estimate the derivatives for a point depending on a local neighborhood around that point in a given dimension $\mathbb{R}^M$. Several different neighborhood criteria can be applied in approximating partial derivatives locally, such as kernel density estimation [43] and geographically weighted regression [7], which result in different estimates depending on the local density of the points. Here, we consider two common neighborhood methods. Let $N(\mathbf{x}) = \{\mathbf{x}_1, \ldots \mathbf{x}_k\}$ be the neighborhood of a point $\mathbf{x}$ in Euclidean space $\mathbb{R}^M$.

One neighborhood, the $R-$ball, is obtained by using a fixed radius R around each data point. A point $\mathbf{x}_i$ is a neighbor of the target point $\mathbf{x}_0$ if $d(\mathbf{x}_i, \mathbf{x}_0) < R$. Another common neighborhood is the $k-$nearest neighbor graph (KNN), where a point is a neighbor of the target point $\mathbf{x}_0$ if it is among the k nearest neighbors other than $\mathbf{x}_0$ itself.

These two criteria provide the user an *adjustable kernel* to accommodate datasets that have a non-uniform density in the



(a) Radius (b) Num of Neighbors

Fig. 11. Different neighborhood kernels and sizes.

projection space. In general, a fixed radius is useful when we are interested in finding smooth flow signatures that explain the sensitivity of a given variable with respect to another. On the other hand, a fixed number of neighbors is more suitable when trying to discriminate the trends between regions of points of disparate density or that are separated in the selected subspace.

Local Density. In addition to estimating the derivative, the local neighborhoods are estimates of the local density of points at any given region in a high dimensional space $\mathbb{R}^M$. For the case of a fixed radius, we can estimate the density as Eq. 10 where $|N(\mathbf{x}_0)|$ is the cardinality of set $N(\mathbf{x}_0)$:

$$\delta(\mathbf{x_0}) \approx \frac{|N(\mathbf{x}_0)|}{R^M} \tag{10}$$

For the case of a fixed number of neighbors:

$$\delta(\mathbf{x_0}) \approx \frac{K}{\max_{\mathbf{x}\in N(\mathbf{x}_0)} d(\mathbf{x},\mathbf{x}_0)^M} \tag{11}$$

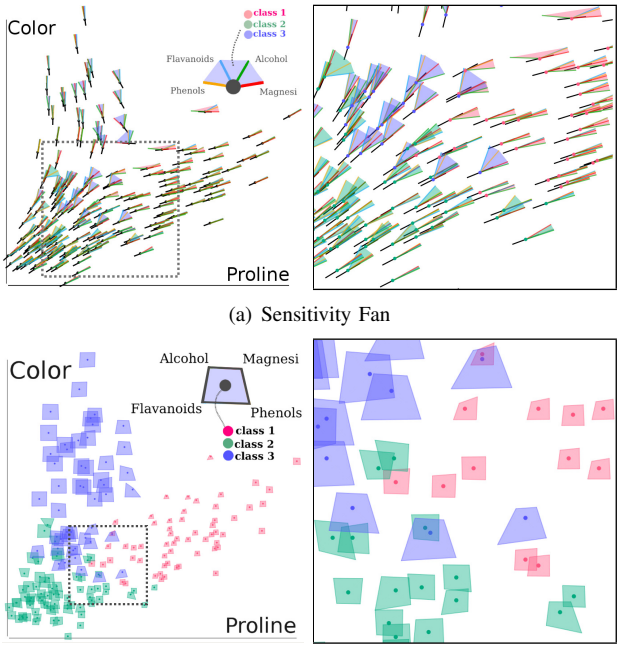
Fig. 11 shows the effect of the neighborhood parameters in the estimated sensitivity lines. It also depicts the estimated local density by coloring each sensitivity line using a warm-to-cool color map, where hot colors highlight locally dense areas while cold colors indicate sparse neighborhoods.

Density estimates may also be useful for decimating sensitivity lines when data points clutter or they appear with varying density. While this may be a requirement for large data sets, existing techniques can be easily adapted to control the visibility and density of sensitivity lines.

5.2 Visual Encoding of Sensitivity

After the steps of subspace selection, differentiation and projection are completed, we can encode the resulting sensitivity in the 2D scatterplot.

One of the goals of the proposed visual glyphs is to let users *visually* separate regions of the high dimensional space from simpler projections. As we shall see, each new glyph exposes dependencies in the data that are not visible from a 2D projection, while retaining the scalability and familiarity of 2D interaction.



(a) Sensitivity Fan

(b) Sensitivity Star Glyphs

Fig. 12. Plot of color vs. proline using sensitivity fans and star glyphs. In this case, star glyphs are effective at visually segmenting the data based on the size and shape of glyphs, into the ground-truth classes, encoded by color.

5.2.1 Sensitivity Lines

In its simplest form, the GSS is formed by drawing unit length lines centered at each data point in the direction of the derivative (u,v). Depending on which dimensions we choose to compute the derivative, the resulting lines will be different. Note that, while sensitivity lines do not cross in a FBS, that is not the case for a GSS, and two points in a vicinity may exhibit very different sensitivities, indicating the effect of a hidden variable.

Sensitivity lines are also limited to a single differentiating (or reference) variable. To understand all sources of sensitivity, one must explore different combinations of subspaces and reference variables, but this is prohibitive to compute or visualize in its entirety. Instead, we consider sensitivities of different subspaces as extra variables that we use to populate 2D glyphs. Two of these, which we call the *sensitivity fan* and the *sensitivity star glyph*, encode the direction and magnitude of multiple sensitivities simultaneously in a single plot.

5.2.2 Sensitivity Fan

The sensitivity fan is a collection of multiple sensitivity lines estimated in subspaces $\{X,Y,Z_i\}$, where X and Y are the projection variables in the scatterplot and Z_i represents the other different dimensions. It is a simplified visualization to overlay and to compare multiple GSSs on the same X-Y projection. To improve readability, instead of showing lines crossing on each data point, we only show half of the sensitivity line for each dimension.

Given a set of k variables $Z = \{Z_1, Z_2, ..., Z_k\}$, the sensitivity fan on a point $\mathbf{x}$ is formed by lines:

$$(u_i(\mathbf{x}), v_i(\mathbf{x})) = \frac{\partial \Pi_i \mathbf{x}}{\partial x} \quad (12)$$

where $\Pi_i : \mathbb{R}^N \mapsto \mathbb{R}^3$, is a projection transformation from the N-dimensional space to the space formed by variables X, Y, Z_i.

Each of these fan lines are constructed by a line segment from $\mathbf{x} = (x, y)$ to $(x + \delta u_i(\mathbf{x}), y + \delta v_i(\mathbf{x}))$ scaled by a given constant factor δ, so that the slope of the fan line denotes the direction of the sensitivity. Lines can be colored differently to enable visual comparison between the different subspaces. We connect adjacent fan lines with a polyline, interpolating the colors associated with each variable to highlight the difference between fan lines and to ease reading fans overlapping each other in a local region. The color of the fans can be turned off using the user interface. An example of sensitivity fans is shown in Fig. 12 for the wine data set.

5.2.3 Sensitivity Star Glyphs

A sensitivity star glyph is similar to a sensitivity fan, but it highlights the *magnitude* of the sensitivities instead. In a sensitivity fan, each line is normalized independently so that the lines have the same length in the plot. This makes it difficult to compare across different data dimensions. Instead, one can consider the magnitude of the sensitivity as an extra dimension and use these dimensions to define a signature shape around each point. This shape is a star glyph and is built as a polygon, where each vertex is placed uniformly around the data point at a distance proportional to the magnitude of sensitivity in each subspace. We follow the design of the radar chart [17] and the star glyph [44].

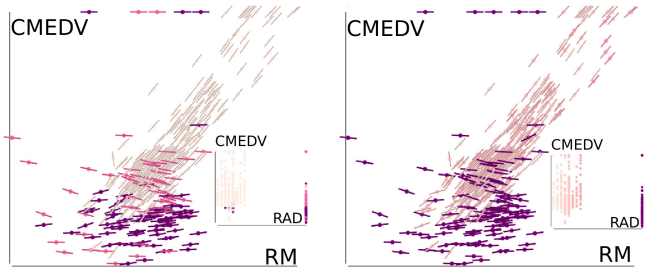
For each variable Z_i, $i \in \{1, \ldots, K\}$ considered in the multi-dimensional sensitivities, we define K directions $\mathbf{d}$ uniformly around a point, and place a vertex at $\mathbf{x} + |(u_i(\mathbf{x}), v_i(\mathbf{x}))|\mathbf{d}$, where $|(u_i(\mathbf{x}), v_i(\mathbf{x}))|$ denotes the magnitude of the sensitivity in the subspace (X, Y, Z_i). An example of sensitivity star glyphs is in Fig. 12.

5.3 Selection and Clustering

An important issue with scatterplots is the selection of meaningful groups. Selection consists of assigning a class to a subset of the points based on a containment or proximity function. In traditional scatterplots, it can be containment in a user-dragged rectangular region, or proximity to an arbitrary *brushing* region. In this paper, we show that the use of sensitivities, especially along different subspaces, provides the user with better hints of the shape of the multi-dimensional space, and thus improves the selection of meaningful groups. We can now select and cluster points in terms of two new similarity metrics, described in the following sections.

5.3.1 Trend Similarity

In general, it is natural to think that nearby points with similar sensitivities are likely to be grouped together, indicating that they are also nearby in the high-dimensional space. Therefore, one can consider a similarity measure as the Euclidean distance between tuples (x, y, u, v) formed by the projection in the 2D space and their respective projected sensitivities. Selection can be defined as proximity in this space. In practice, a user simply selects points in terms of proximity in the 2D space, then those points with a sensitivity outside a user-specified range are removed from the selection.



(a) Unsupervised clustering (b) Classification by RAD

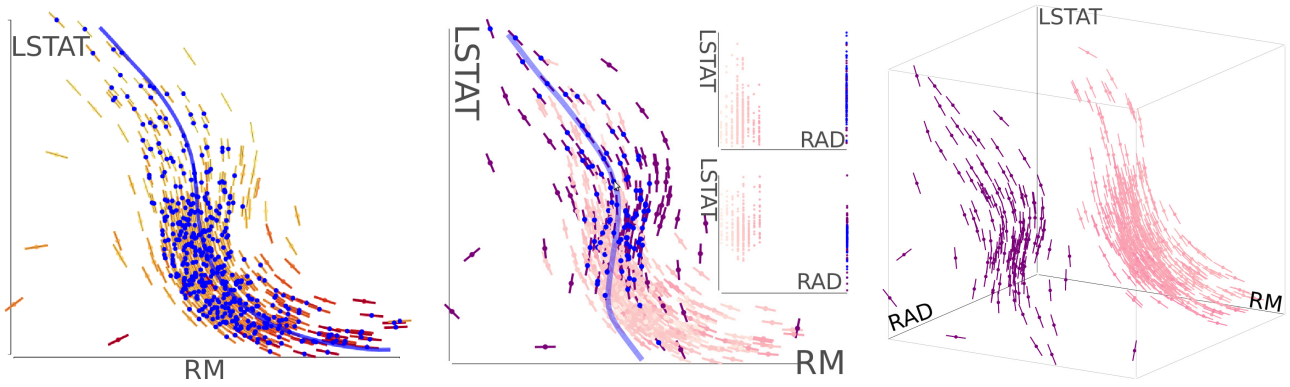
Fig. 13. (a) Clustering of median housing price (CMEDV) as a function of number of rooms (RM). While in a 2D scatterplot the groups are not separable, the automatic clustering of sensitivity in the subspace (CMEDV,RM,RAD) picks up both visually and analytically the difference in trend, exposing two groups (b) These groups can be evident if we classify by variable RAD, which is hidden in the 2D plot (only shown in insets)

More interesting is the generalization of the selection process to automatically cluster the data points. We automatically assign classes to points that share similarity depending on both their location and derivative. We used the k-means algorithm to classify points in the four dimensional space (x, y, u, v) formed by data points and their derivatives. An example is shown in Fig. 13(a) for the Boston housing data set [23]. Here we consider the subspace that shows the relationship between the average number of rooms in a house (RM) and the median housing price (CMEDV). In a traditional scatterplot, it may not be obvious that this relationship is not entirely monotonically increasing and there are two main groups, which could be exposed by looking at a third dimension (RAD), which indicates the accessibility of a home to radial highways. A GSS highlights these groups visually, and automatic clustering of the sensitivities validate the visual grouping. Compare to Fig. 13(b), which color codes the data points based on the hidden variable RAD. While the automatic clustering in (a) is not perfect, it is effective at highlighting two main groups, one formed by the points in light pink exhibiting a strong linear relationship between RM and CMEDV and another formed by bright pink and purple points.

The implications of this are important, since it shows that clustering in a the 4D subspace formed by points and sensitivity may be as accurate as clustering in a possibly high-dimensional space.

5.3.2 Distance to Streamline

Another way to select points is to follow the structure of the data. In our previous work [10], we showed that FBS give rise to streamlines as a metaphor for showing global trends. Streamlines are computed in 2D by integrating the sensitivity lines from a given seed point $\mathbf{x}$, using Runge-Kutta methods. Because we deal with scattered points, FBS use



(a) FBS (b) GSS (c) 3D Scatterplot

Fig. 14. Streamline selection by (a) 2D distance on (RM, LSTAT); (b) 3D view shows pink and purple trends are differentiated by RAD; (c) GSS of (RM, LSTAT, RAD).

scattered interpolation to predict the value and derivative at each sampled point, using similar kernels to the ones used in sensitivity estimation.

Unlike our previous work, we compute streamlines in the selected subspace, a possibly high-dimensional space. In addition, we do not rely on multiple streamlines to encode trends. We have noticed that predicting flow far away from sample points may create trends that extend well beyond the bounding hyper-volume of the scattered points and thus may be misleading. Instead, we only estimate streamlines for individual seed points as selected to the user to define a guide for selection. Computing and drawing a single streamline of a data point of interest is real-time and highly interactive. In the worst case, its computational time scales linearly with the number of data points, but with the aid of spatial data structures such as kd-trees, it can be done in logarithmic time.

Now we can define a selection in a more "data-aligned" manner: instead of Euclidean distance to a single point, we compute the shortest distance from a point to the selected line in the selected subspace. A point is said to be in the selection if its shortest distance to the streamline is less than a given threshold. This requires two interactive parameters: the streamline seed, often selected as the user hovers over data points, and the distance to the streamline, or width of the selection.

We show an example in Fig. 14 using the Boston data set [23], analyzing the relationship between LSTAT (the percentage of the lower status of the population) and RM (average number of rooms per dwelling). Fig. 14(a) shows the selection using FBS, where the selected streamline is shown as a solid blue line, and selected points in blue. Notice how the selection is not axis-oriented but rather feature oriented.

Fig. 14 we show selection on a GSS using the subspace formed by LSTAT, RM and RAD. The sensitivity lines hints at us of the presence of two distinct regions, judging by the trends of the sensitivity lines. When we select points by streamline, this high-dimensional curve (projected onto 2D) is aligned with the feature formed by purple points (with a high RAD), allowing us to select points in that region of hyper-space. In contrast to FBS, selection in a GSS plot helps us detect a relationship that could only be evident in a 3D space, as shown in Fig. 14(c). In traditional scatterplots, it would not be possible to discern the data points in this region without access to a third dimension. Our generalized sensitivity scatterplot includes the extra dimension implicitly in the view, which becomes more space-efficient and demands less time when navigating multi-dimensional spaces.

6 Results

We now show examples of GSS in the analysis and visualization of multi-dimensional data sets. We aim to show that augmenting scatterplots with sensitivity information helps analysts identify trends and groups in 2D with little need for interaction, while these may only be evident in traditional scatterplots with access to multiple dimensions or after tedious and time-consuming user interaction.

6.1 Wine Data Set

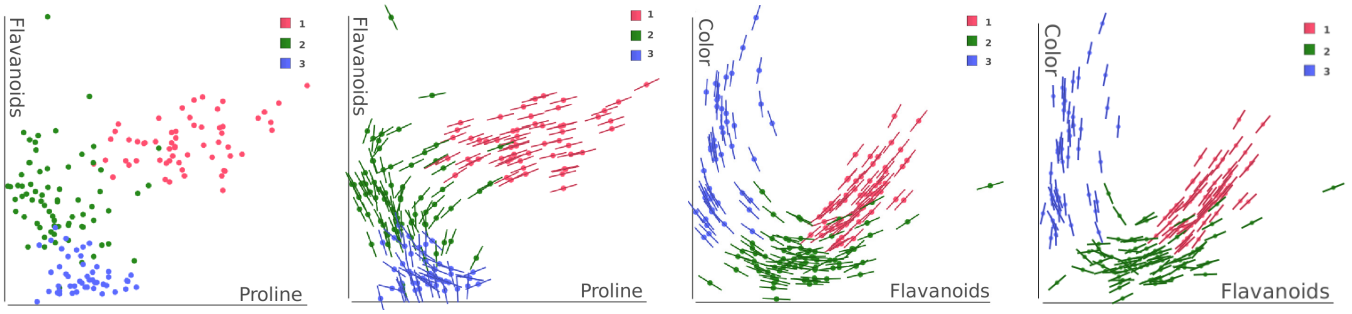
As described in Sec.1.1, the wine data set [47] comprises 13 variables of 178 observations of the chemical composition of wines growing in Italy, and a classification of these wines in three classes. One of the interesting questions regarding this data set is whether it is possible to understand which variables are key in explaining these classes.

For a traditional scatterplot in Fig. 15(a), the classes may be overlapping or their boundaries may not be clearly defined. We notice that no single 2D projection can produce a perfect separation of these classes. Therefore, we turn to our GSSs to examine if sensitivity information can help us to better understand this data set.

Fig. 15(b-c) shows the GSSs of the selected subspace: (P,C,F), projected on the 2D subspace (P,F) (P: the percentage of proline; C: the color intensity; F: the percentage of flavanoids).

Although on the projection (P,F) in Fig. 15(b) we observe two distinct trends (one mostly composed by class 1 points in red, and the other formed mostly by classes 2 and 3) , these do not perfectly segment the data into the three classes of wines.

Fig. 15(c) shows a different projection (F,C) from the same selected subspace. We see that the sensitivity lines clearly delineate two different trends: one for class 1 wines with a linear relationship between color and concentration of flavanoids, and another non-linear relationship formed by the other two classes.



(a) scatterplot (b) GSS of a subspace (P, F, C) (c) GSS on a different projection (d) GSS of a 8-D subspace

Fig. 15. GSSs of the wine dataset. (a) class boundaries are not clear; (b) provides more separation cues; (c) class 1 exhibits a linear relationship distinguishable from others; (d) shows sensitivities of an 8-D subspace that class 2 and 3 are distinguishable on trends.

To help us identify differences between classes 2 and 3 wines we select a higher dimensional subspace for sensitivity, as shown in Fig. 15(d). In this case we select the subspace (P,C,F,A,H,M,O,Pr) (A: Alcohol; H: Hue; M: Magnesi; O: OD280; and Pr: Proanthocyanin). We observe that class 2 and 3 wines, which could not be segmented in (c) are now easy to segment in terms of their sensitivity in (d), judging by the sudden change in the orientation of the lines (minus a few exceptions). From Fig. 15(c-d) we see that sensitivity is a useful tool for visually segmenting data without the need for complex multi-dimensional interactions.

Besides the ability to visualize sensitivities from a selected subspace, we would still need a mechanism to explore sensitivities in a higher dimensional space systematically. Therefore we turn to sensitivity fans and the star glyphs to summarize the sensitivity parameters along multiple dimensions. Fig. 12(a-b) showed the projected subspace of *proline* vs. *color* using sensitivity glyphs to summarize four sensitivities simultaneously, namely the concentration of *magnesium* (red), *phenols* (yellow), *alcohol* (green) and *flavanoids*(blue). In Fig. 12(a), we see regions where the *color-proline* relationship has little sensitivity to other dimensions, judging by how "closed" the fan is. The more closed a fan is, i.e., appearing more like a single line segment, the less impact have the other variables into explaining the 2D relationship. This is the case for those data points with a high *proline* (rightmost points), or high *color* intensity (topmost points). They both have closed fans. However, in the region in the middle of the plot we observe larger discrepancies, depicted as larger or "open" fans. These fans indicate that color, as a function of *proline*, is more sensitive to the concentration of *flavanoids* (blue) and *phenols* (yellow) than to the concentration of *magnesium* (red).

Then we explore the data using sensitivity star glyphs in Fig. 12(b) and identify three distinct groups by the distinct shapes of the quads spanned by the four query variables. In fact, we can now segment the data into their respective classes by looking at the relative sizes of the star glyphs, which indicate the magnitude of the sensitivity. For example, class 1 wines appear less sensitive to the four query variables (pink quads), while class 3 wines exhibit more sensitivity (blue quads). We also observed that the class 3 wines highlighted in the black rectangular region, where the three classes overlap, exhibit more sensitivity to the concentration of *flavanoids* and *phenols*. However the class 3 wines in the topmost part of the scatterplot exhibit a rather homogeneous sensitivity along all four query dimensions.

6.2 Automobile MPG

The Automobile mpg data set concerns city-cycle fuel consumption in miles per gallon for a number of automobiles [36]. It contains 398 records for different cars made between 1970 and 1982, with eight attributes, five of which – *miles-per-gallon* (*MPG*), *weight*, *acceleration*, *horsepower* and *displacement* –are continuous. We use the predicted variable MPG as the output variable and investigate its relationship with other dependent variables. To better understand the relationship between these variables, we used the number of *cylinders*, a discrete variable, as a classification variable.

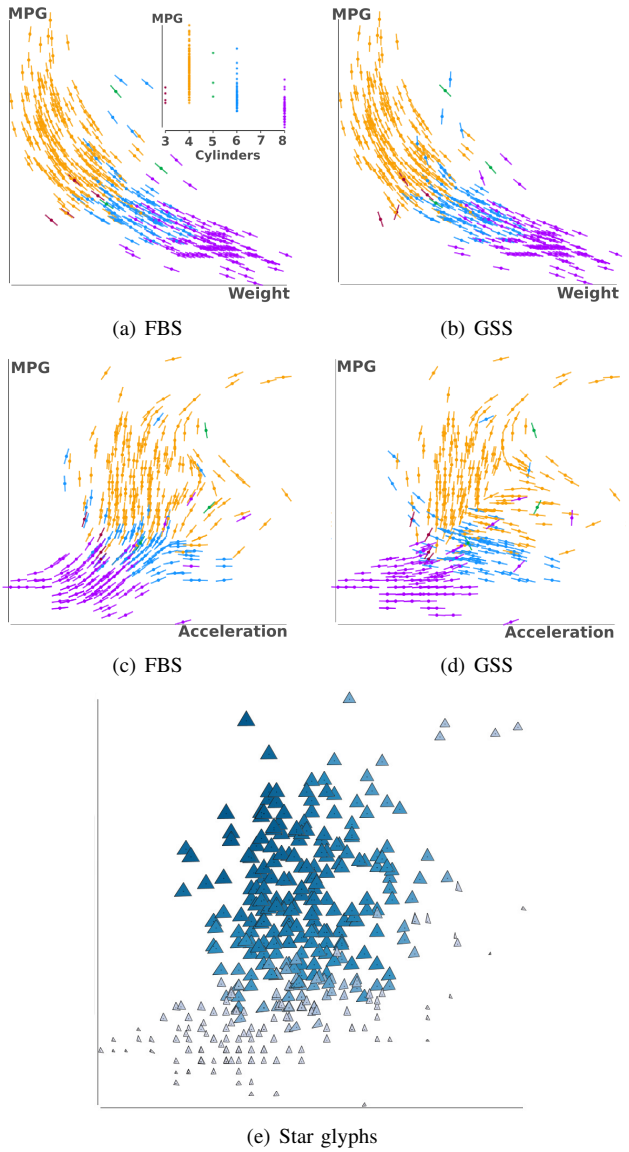
Figs. 16(a) and (c) show the FBS of the 2D spaces formed by *MPG* and *weight*, and *MPG* and *acceleration*, respectively. In contrast, Figs. 16(b) and (d) show the GSS of the same projections, using sensitivities formed by the 3D subspace of *MPG*, number of *cylinders*, and either *weight* or *acceleration*.

From these plots, we observe the following relationships:

1. *MPG* vs. *Weight*. Fig. 16(a-b) shows that there is an inverse relationship between *weight* and *MPG*. The fact that this relationship does not seem to change when considering the 3D subspace (considering number of *cylinders*) shows that the relationship between *MPG* and *weight* is insensitive to the number of *cylinders* in the car, and validates the intuition that heavier cars consume more fuel.

2. *MPG* vs. *Acceleration*. The 2D scatterplot in Fig. 16(c) suggests little correlation between the two, though some positive relationship emerges locally, as suggested by the flow lines. However, the GSS in Fig. 16(d) suggests differently: for 8-cylinder cars (purple), the correlation is mostly positive, while for 6-cylinder (blue) cars they exhibit an inverse relationship.

We also analysed the magnitude of sensitivity using star glyphs, as shown in Fig. 16(e). This allows us to show the sensitivity of multiple variables (*horsepower*, *displacement* and *weight*) in a single plot. To better highlight the differences between points, we color coded the triangles by area. High *MPG* cars exhibit larger sensitivity than those with low *MPG*, as evidenced by the large size of the respective triangles. The relationship between *acceleration* and *MPG* seems less uncertain for 8-cylinder cars, as suggested by the small triangles



(a) FBS (b) GSS

(c) FBS (d) GSS

(e) Star glyphs

Fig. 16. (b) and (d) show the GSSs of (a) and (c) with the extra dimension *cylinders*. (d) indicates the trends between *MPG* and *Acceleration* is more different for the 6-cylinder cars (in blue) than in (b). (e) highlights high sensitivity data.

at the bottom of the plot. The 4-cylinder cars have a larger sensitivity, which also explains why there seems to be little correlation between these two variables, suggesting there is more variability in cars with fewer cylinders.

7 Conclusions and Future Work

Although simple in nature, the scatterplot remains a powerful visual representation of bivariate data. It provides visual cues about the relationship between two variables, in terms of proximity of data points, continuity and the saliency of outliers. In this paper, we present a visual augmentation of scatterplots that introduces sensitivity information. As we have shown with examples, our generalized sensitivity scatterplot (GSS) retains the perceptual benefits of a scatterplot that it does not hide any information that could be extracted from the original scatterplot, while it also introduces new visual cues that provide better insight into the relationship between two data variables. First, the orientation cues provided by the flow lines give an idea of the local trend of data and a sense of continuity not graspable from the unaugmented scatterplot. Second, we have shown that these cues become useful when flow lines are obtained from a higher dimensional space than the subspace represented in the plot, since coherent groups of trend lines are perceptually interpreted as smooth regions that can only be visualized in a higher dimensional space. These properties make the GSS a more efficient visualization technique for exploring multi-dimensional data since it alleviates the need for time-consuming change of coordinates and re-projections.

One can understand the flow lines as gradient lines from an underlying density function estimated from the data points, and that is why a GSS is effective as an accurate depiction of both the function values and local derivative. Nonetheless, we have also shown the benefits of extending this idea to provide more abstract plots, where glyphs are used to summarize multiple sensitivities. This is the case of the sensitivity fan and sensitivity star glyph, which summarize the orientation and magnitude of the sensitivity, respectively. As shown in our examples, the shapes of these 2D glyphs showing multiple three-dimensional sensitivity derivative are useful to group data points in a higher dimensional space, which may not be possible in a low dimensional projection such as the scatterplot itself. This flexibility makes the GSS useful for both regression tasks – such as explaining the relationship between fuel efficiency, horse power and weight in the automobile example – and classification problems, such as the ability to predict the number of cylinders in a car given its different attributes.

We would like to explore the analysis of GSS in a more general setting where ground truth data is not known, and compile testimonies from users regarding its use as a visual analytics tool. We believe this paper provides the technical foundation that will spark user studies that show to what extent users can interpret multi-dimensional data using a GSS and draw correct conclusions about their data. Our experiment has resulted in an unprecedented data set of over six thousand curves and one hundred functions that we will make publicly available to foster further research about the benefits of scatterplots, trend lines and function fitting.

Acknowledgment

This research was sponsored in part by HP Labs, Northrop Grumman Corporation, and the U.S. National Science Foundation through grants CCF-1025269, CCF-0811422, and CCF-0808896.

References

[1] L. Arriola and J. Hyman. Being Sensitive to Uncertainty. *Computing in Science & Engineering*, 9(2):10–20, 2007.

[2] S. Bachthaler and D. Weiskopf. Continuous scatterplots. *IEEE Trans. Vis. & Computer Graphics*, 14(6):1428–35, 2008.

[3] S. Barlowe, T. Zhang, Y. Liu, J. Yang, and D. Jacobs. Multivariate Visual Explanation for High Dimensional Datasets. In *VAST*, pages 147–154, 2008.

[4] W. Berger, H. Piringer, P. Filzmoser, and E. Gröller. Uncertainty-Aware Exploration of Continuous Parameter Spaces Using Multivariate Prediction. *Computer Graphics Forum*, 30(3):911–920, 2011.

[5] P. Berkhin. Survey of Clustering Data Mining Techniques. In *Grouping Multidimensional Data*, pages 25–71. Springer, 2006.

[6] G. E. P. Box and N. R. Draper. *Empirical Model-Building & Response Surfaces*. Wiley, 1987.

[7] C. Brunsdon, S. Fotheringham, and M. Charlton. Geographically Weighted Regression. *Journal of Royal Statistical Society*, 47(3):431–443, 1998.

[8] D. G. Cacuci. *Sensitivity & Uncertainty Analysis: Theory*. CRC Press, 2003.

[9] K. Chan, A. Saltelli, and S. Tarantola. Sensitivity Analysis of Model Output. In *Proc. of the 29th Conf. on Winter Simulation*, pages 261–268, 1997.

[10] Y.-H. Chan, C. D. Correa, and K.-L. Ma. Flow-based Scatterplots for Sensitivity Analysis. In *IEEE Symposium on VAST*, pages 43–50, 2010.

[11] M. Chau, R. Cheng, B. Kao, and J. Ng. Uncertain Data Mining: An Example in Clustering Location Data. In *KDD*, volume 3918, pages 199– –204. Springer, 2006.

[12] C. Collins, G. Penn, and S. Carpendale. Bubble Sets: Revealing Set Relations with Isocontours over Existing Visualizations. *IEEE Trans. Vis. & Computer Graphics*, 15(6):1009–16, 2009.

[13] G. Cormode and A. McGregor. Approximation algorithms for Clustering Uncertain Data. In *ACM Symposium on Principles of Database Systems*, page 191, 2008.

[14] C. D. Correa, Y.-H. Chan, and K.-L. Ma. A Framework for Uncertainty-Aware Visual Analytics. In *IEEE Symposium on VAST*, page 191, 2009.

[15] N. R. Draper and H. Smith. *Applied Regression Analysis*. Wiley, 3 edition, 1998.

[16] N. Elmqvist, P. Dragicevic, and J.-D. Fekete. Rolling the Dice: Multidimensional Visual Exploration using Scatterplot Matrix Navigation. *IEEE Trans. Vis. & Computer Graphics*, 14(6):1141–8, 2008.

[17] T. G. Eschenbach. Spiderplots versus Tornado Diagrams for Sensitivity Analysis. *Interfaces*, 22(6):40–46, 1992.

[18] D. Feng, L. Kwock, Y. Lee, and M. Taylor. Matching Visual Saliency to Confidence in Plots of Uncertain Data. *IEEE Trans Vis. & Computer Graph*, 16(6):980–989, 2010.

[19] H. C. Frey and S. R. Patil. Identification & Review of Sensitivity Analysis Methods. *Risk Analysis*, 22(3):553–78, 2002.

[20] Y.-H. Fua, M. O. Ward, and E. A. Rundensteiner. Structure-based brushes: A mechanism for navigating hierarchically organized data and information spaces. *IEEE Transactions on Visualization and Computer Graphics*, 6(2):150–159, Apr. 2000.

[21] A. Griewank and A. Walther. *Evaluating Derivatives: Principles & Techniques of Algorithmic Differentiation*. SIAM, 2008.

[22] Z. Guo, O. Ward, A. Rundensteiner, and C. Ruiz. Pointwise Local Pattern Exploration for Sensitivity Analysis. In *VAST*, page 438, 2011.

[23] D. Harrison and D. Rubinfeld. Boston Neighborhood Housing Price dataset (BNHP). http://lib.stat.cmu.edu/S/Harrell/data/descriptions/boston.html. [Online; accessed 01-Sep-2011].

[24] T. Hastie and R. Tibshirani. *Generalized Additive Models*. Chapman & Hall/CRC, 1990.

[25] J. Heinrich, S. Bachthaler, and D. Weiskopf. Progressive Splatting of Continuous Scatterplots & Parallel Coordinates. *Computer Graphics Forum*, 30(3), 2011.

[26] J. Helton, J. Johnson, C. Sallaberry, and C. Storlie. Survey of Sampling-based Methods for Uncertainty & Sensitivity Analysis. *Reliability Engineering & System Safety*, 91(10-11), 2006.

[27] R. L. Iman and J. C. Helton. An Investigation of Uncertainty & Sensitivity Analysis Techniques for Computer Models. *Risk Analysis*, 8(1):71–90, 1988.

[28] M. Jansen. Analysis of Variance Designs for Model Output. *Computer Physics Communications*, 117(1-2):35–43, 1999.

[29] D. H. 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.

[30] D. A. Keim, M. C. Hao, U. Dayal, H. Janetzko, and P. Bak. Generalized Scatter Plots. *Information Visualization*, 9:301–311, 2009.

[31] D. Kurowicka and R. M. Cooke. *Uncertainty Analysis with High Dimensional Dependence Modelling*. Wiley, 2006.

[32] A. R. Martin and M. O. Ward. High dimensional brushing for interactive exploration of multivariate data. In *Proceedings of the 6th conference on Visualization '95*, VIS '95, pages 271–, Washington, DC, USA, 1995. IEEE Computer Society.

[33] R. McGill, J. Tukey, and W. Larsen. Variations of Box Plots. *American Statistician*, 32(1):12–16, 1978.

[34] A. Moore, J. Schneider, and K. Deng. Efficient Locally Weighted Polynomial Regression Predictions. In *ICML*, pages 236–244, 1997.

[35] K. Potter, J. Kniss, R. Riesenfeld, and C. R. Johnson. Visualizing Summary Statistics and Uncertainty. *Computer Graphics Forum*, 29(3):823–831, 2010.

[36] Quinlan, R. Auto MPG Data Set. http://archive.ics.uci.edu/ml/datasets/Auto+MPG. [Online; accessed 01-Sep-2011].

[37] J. Shlens. A Tutorial on Principal Component Analysis. *Measurement*, 51(10003):52, 2005.

[38] B. Shneiderman and A. Aris. Network Visualization by Semantic Substrates. *IEEE Trans. Vis. & Computer Graphics*, 12(5):733–40, 2006.

[39] V. Smidl and A. Quinn. On Bayesian Principal Component Analysis. *Computational Statistics & Data Analysis*, 51(9):4101–4123, 2007.

[40] I. M. Sobolá. Global Sensitivity Indices for Nonlinear Mathematical Models & Their Monte Carlo Estimates. *Mathematics & Computers in Simulation*, 55(1-3):271–280, 2001.

[41] Y. Tanaka. Recent advance in Sensitivity Analysis in Multivariate statistical methods. *Computational Statistics*, 7(1):1–25, 1994.

[42] S. K. Thompson. *Sampling*. Wiley, 2 edition, 2002.

[43] M. P. Wand and M. C. Jones. Comparison of Smoothing Parameterizations in Bivariate Kernel Density Estimation. *Journal of the American Statistical Association*, 88(422):520–528, 2012.

[44] M. O. Ward. A Taxonomy of Glyph Placement Strategies for Multidimensional Data Visualization. *Information Visualization*, 1(1):194–210, 2002.

[45] E. W. Weisstein. Least Squares Fitting. http://mathworld.wolfram.com/LeastSquaresFitting.html. [Online; accessed 01-Sep-2011].

[46] E. W. Weisstein. Least Squares Fitting – Perpendicular Offsets. http://mathworld.wolfram.com/LeastSquaresFittingPerpendicularOffsets.html. [Online; accessed 01-Sep-2011].

[47] Wine Data Set. http://archive.ics.uci.edu/ml/datasets/Wine.

[48] Y. Yamanishi and Y. Tanaka. Sensitivity Analysis in Functional Principal Component Analysis. *Computational Statistics*, 20(2):311–326, 2005.

[49] D. Yang, E. Rundensteiner, and M. Ward. Analysis Guided Visual Exploration of Multivariate Data. In *IEEE Symposium on VAST*, pages 83–90, 2007.

Yu-Hsuan Chan is a PhD candidate in Computer Science at UC Davis. She received her BS and MS degrees in Industrial Engineering and Engineering Management from National Tsing Hua Univ. in Taiwan in 2005 and 2007 respectively. Her research interests include uncertainty-aware visual analytics, sensitivity analysis for multidimensional data, and Social Network Analysis.

Carlos D. Correa is a computer scientist at the Center for Applied Scientific Computing at Lawrence Livermore National Lab. His research interests include scientific and information visualization, uncertainty analysis and visualization, and computer graphics. Before joining LLNL, he worked as a postdoctoral researcher at UC Davis. He earned an MSc and a PhD in Electrical and Computer Engineering from Rutgers Univ. in 2003 and 2007, respectively. He holds a BSs in Computer Science from EAFIT Univ., Colombia.

Kwan-Liu Ma is a professor of Computer Science at the University of California, Davis. He leads the VIDi research group and directs the UC Davis Center for Visualization. Professor Ma received his PhD degree in Computer Science from the University of Utah in 1993. He was a research scientist at the ICASE/NASA LaRC before joining UC Davis in 1999. His research interests include data visualization, visual analytics, and high-performance computing. Professor Ma is an IEEE Fellow, and a founder of the IEEE Pacific Visualization Symposium and IEEE Symposium on Large Data Analysis and Visualization. He can be reached via ma@cs.ucdavis.edu.