SketchPad^{N-D}: An Interface for High-Dimensional Dataset Generation and Editing

Puripant Ruchikachorn[1,2,*] Bing Wang[1,*] Klaus Mueller[1]

[1]Department of Computer Science, Stony Brook University
[2]Chulalongkorn Business School, Chulalongkorn University

ABSTRACT

In order to generate data with known and desired features for high-dimensional data testing, we propose a tool that allows users to generate multivariate data directly within the same interface they would also use to visualize the data. We demonstrate our ideas with two well-established visualization paradigms, one based on the parallel coordinate framework, the other based on scatterplots.

Keywords: Synthetic data generation, multivariate data, parallel coordinates, scatterplot.

Index Terms: H.5.2 [INFORMATION INTERFACES AND PRESENTATION]: User Interfaces — Graphical user interfaces (GUI); I.3.3 [COMPUTER GRAPHICS]: Picture/Image Generation — Display algorithms, Viewing algorithms.

1. INTRODUCTION

High-dimensional data analysis and visualization is useful in many applications and domains. Designing new algorithm and software requires datasets with specific features for testing. However, real datasets are in limited supply and those available often lack the features needed for targeted evaluations. Therefore, we propose an interface for high-dimensional dataset generation.

The proposed SketchPad^{N-D} is tightly integrated with high-dimensional data visualization. Users need not switch back and forth between data generation and visualization tools as they are combined into one interface. This provides better context for later iterations of the data generation process and facilitates a more streamlined workflow. The interface we describe is based on two visualization techniques, namely parallel coordinates and scatterplots. Users can rely solely on one of the two tools, or alternatively start with one tool and use the other to further edit the current data. The workflow for data generation is shown in Figure 1.

2. RELATED WORK

There are a number of automatic data generation methods for specific applications such as software testing based on constraints [1]. Albuquerque et al. [2] recently presented a framework for high-dimensional data generation that is visual and interactive but has two major limitations. Firstly, it allows users to define only univariate and bivariate distributions in scatterplot. Secondly, it simply discards all distributions on a dimension if it has been previously defined.

Sketch-based interfaces have been explored in many applications, such as gestures for entering music notations [3] and solving mathematical problems [4].

Email: {pruchikachor, wang12, mueller}@cs.stonybrook.edu
* These authors contributed equally to this work.

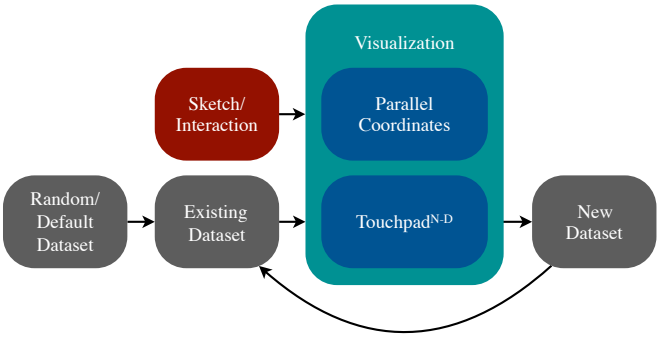

Figure 1: The overview of our workflow.

3. USER INTERFACES

3.1 Sketching on Parallel Coordinates

A sample input sketch and its data generation result are shown in Figure 4(b) to illustrate two available input types.

3.1.1 Probability Density Function (PDF) Sketch

Users can freely draw an arbitrary curve along any axis to specify how data should distribute in the corresponding dimension. This curve is then interpreted as a PDF of the data in that dimension, with associated skewness, kurtosis, and other properties.

The PDF is then sampled to create a discrete cumulative distribution function (CDF) for data generation purpose. Following inverse transform sampling, a data value in a PDF-specified dimension is generated by finding the index of a uniformly distributed random variable in [0, 1) from the discrete CPF. As these PDF sketches are independent to each other, only the data at their own dimensions are updated.

3.1.2 Data Connection Quadrilateral

Users can click between any adjacent pair of axes to add a vertex to a quadrilateral. Four clicks form a shape that can be simple or complex (self-intersecting). Adhering to the common patterns often observed in parallel coordinates, a trapezoid and a bowtie represent a direct and an inverse relationship respectively.

Derived from the generated value for each data sample at a dimension and the input quadrilateral, a temporary distribution for the next (neighboring) dimension is created and used to generate a value for that dimension in this sample. Based on rejection sampling, accepted are only values within the range of their temporary distributions. This step is repeated through all

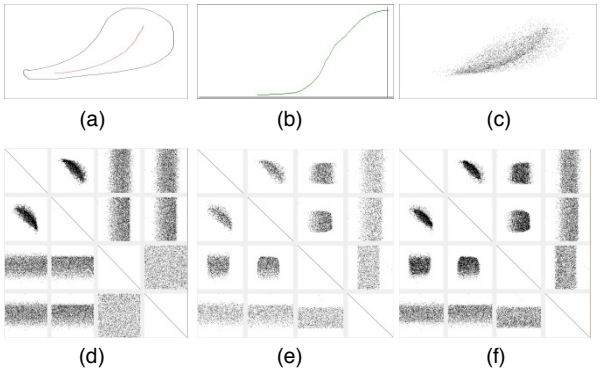

Figure 2: Axis-aligned result.

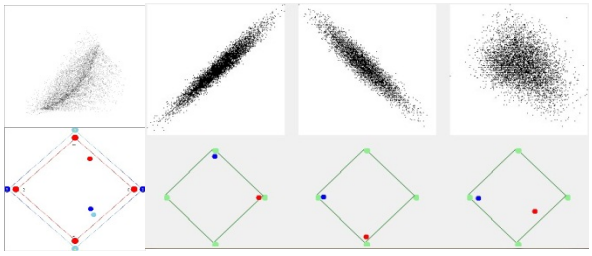

Figure 3: Non-axis aligned result.

dimensions for one sample. Users can also specify the correlation strength for each bivariate distribution.

3.2 Sketching on Scatterplot

3.2.1 Axis-aligned Sketching

The axis-aligned scatterplot based data generating algorithm is as follows. We first initialize points by defining the centerline (where the points are the densest) and the boundary (where no points should go any further) of the point cloud along with the desired distribution from the centerline to the boundary, which forms a probability map. Next, for each point, we use the probability map to pick a coordinate on the selected plane, randomize the value of each of the other dimensions between [0, 1]. Then we do distribution carving by taking any projection and brushing away points in the unwanted places, which will also be removed from all other views. The last step is called repair since erasing points in one view may delete points in other views in undesired placed. Users may use the brushing tool to add those points back. Our system automatically makes sure no points appear in the locations of the views where users erased before. Users may repeat this process until they like what they sketched.

One generated data set is given in figure 2. Figure 2(a) – 2(c) show the drawn shape, the distribution and the initially generated point cloud. Figure 2(d) shows the points in scatterplot matrix. Figure 2(e) and 2(f) describe step 3 and step 4.

3.2.2 Non axis-aligned Sketching

The algorithm for non-axis aligned sketching differs from the axis-aligned case in the following two ways. First of all, we are now defining the probability map on an arbitrary projection plane and thus, when doing backprojection in the second step, we could not simply randomize the value of each coordinate. We therefore for each point drawn, build a linear equation system to relate the coordinates of its projection and its original

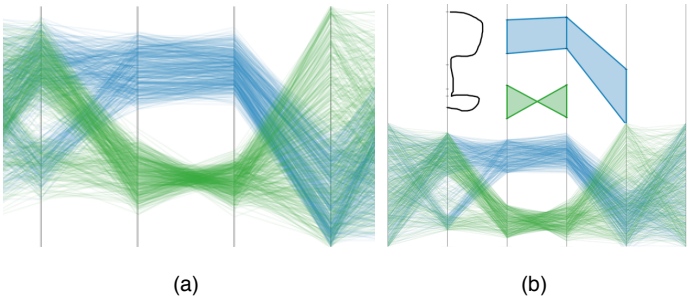

Figure 4: (a) The target result and (b) an example of the result by a user in the user study.

coordinates. Secondly, we could not do the repair step in this schema since all selected projection planes are unlikely perpendicular to each other and it will be contradictory if we remove a point on one plane while keeping it on the rest. Figure 3 shows a 4D structure that has been carved on one projection plane while the rest projections, especially axis-aligned ones, could not display this structure at all.

4. Implementation and User Study

We implemented the parallel coordinates interface in Processing on a 2.4 GHz Intel Core i5 computer with 4 GB of RAM. The user interface for the scatterplot has been implemented in C# and runs on a 2.8 GHz Intel Core i7 computer with 12GB of RAM.

We have also conducted an informal user study to test our interfaces. Eight first-time users were given a brief demonstration of all features of the first user interface, a five-minute tutorial under our guidance, and a task to generate the 6D dataset of 1000 samples with significant features in four dimensions as shown in Figure 4(a). The feedbacks were generally positive. On average, users reset 1.67 times and finished the task in 2 minute and 36 seconds of the last attempt.

5. Conclusion

We presented a new workflow and user interfaces for high-dimensional dataset generation. Our interfaces are fully integrated with two prevailing visualization techniques so users can create a dataset quickly and perform a greater number of tests. At the same time, our interfaces also give users more precise and visual control over the intrinsic attributes of the generated high-dimensional data.

In the future we would also like to support a larger number of dimensions. Here we may employ techniques from subspace clustering and multi-scale zooming that presents different level of details and tools at each scale.

References

[1] R. DeMillo and A. Offutt, “Constraint-Based Automatic Test Data Generation,” IEEE Transactions on Software Engineering, Jan. 1997.

[2] G. Albuquerque, T. Löwe, and M. Magnor, “Synthetic generation of high-dimensional datasets,” IEEE transactions on visualization and computer graphics, vol. 17, no. 12, pp. 2317-2324, Dec. 2011.

[3] A. Forsberg, M. Dieterich, and R. Zeleznik, “The Music Notepad,” in Proceedings of UIST ’98, ACM SIGGRAPH, 1998.

[4] J. LaViola and R. Zeleznik, “MathPad^2: A system for the creation and exploration of mathematical sketches,” ACM Transactions on Graphics, 2004.