Interactive Data Exploration with “Big Data Tukey Plots”

Peter Schretlen, Nathan Kronenfeld, Derek Gray, Jesse McGeachie, Eric Hall,
Daniel Cheng, Nicole Covello, William Wright *

Oculus Info Inc.

Abstract

Before testing hypotheses, confirmatory data analysis benefits from first examining the data to suggest hypotheses to be tested. This is known as exploratory data analysis (EDA). Our goal is to develop new automated tools to produce effective raw data characterization on extremely large datasets. This paper reports on the development of a web based interactive scatter plot prototype that uses tile-based rendering similar to geographic maps and interaction paradigms. Geographic maps share much in common with scatter plots. Both feature continuous data along two dimensions, use of layering and legends, axes and scales. Web delivery of maps using tiled rendering has benefitted from years of work. With widespread use, map interactions have become familiar and make exploration of an abstract large data space easy, even enjoyable. Using similar techniques, our big data tile rendering for scatter plots provides interactive data exploration with continuous zooming on hundreds of millions of points.

Keywords: Multiresolution Techniques, Scalability Issues, Interaction Design, Zooming and Navigation Techniques.

1 Background

There are several classes of users that need to have an overview of a new dataset for exploration and understanding. Data scientists need to design and apply correct analytics. Visualization designers must select appropriate visual metaphors and interactions. Analysts need an understanding of the data in order to know which datasets to select and which tools to use to approach a given analytic question. However, data can be in any format and “big data” is often too large to determine what analytics to use without sampling.

Before testing statistical hypotheses, confirmatory data analysis benefits from first examining the data to suggest hypotheses to be tested. This is exploratory data analysis (EDA) [1]. For big data, the exploratory process needs to be automated to operate at scale. Current tools, like Tableau, R, Python, have powerful libraries for data manipulation and analysis but operate on subsets of big data.

For the purpose of exploratory data analysis our goal is to develop new automated tools to produce effective raw data characterization on large datasets. There should be minimal need for user input, these capabilities should be built using freely available and open source tools and platforms, and model building (e.g. regressions, dendograms, k-means clustering) is deferred to the subsequent analytical question stage. Methods that we have chosen to visually communicate big data characteristics include:

- Summary statistics of the data (e.g. counts, size, format, schema, metadata, quartiles, mean, median, top frequencies);
- Cross plots of every attribute against every attribute, except in cases of columns containing only unique entries or containing only one grouping (i.e. the same value is seen in the entire table);
- Box plots of any attribute that contains numerical values;
- Geographic plots of any attributes that contain location data;
- Frequency histograms of all attributes, except columns containing only unique entries or only one grouping; and
- Quality assessment (e.g. missing, outliers, poor data entry).

2 Approach

To assess the feasibility and utility of using tile-based rendering for big data scatter plots, to identify usability and user experience issues, and to develop requirements for further implementation, we are experimenting with a variety of datasets, including:

- Kiva Microfinance (500K points, 5.9 GB)
- Bitcoin (37M points, 3.6 GB)
- AIS vessel data (112M points, 29 GB)
- Twitter (292M points, 146 GB)
- Trace route data (961M points, 157 GB)

Challenges include determining how to modify web mapping tools to do what is expected in a scatter plot (e.g. axes, labeling, etc.), which Javascript libraries are best suited for this purpose, and how data requirements differ between maps and scatter plots and how to address these differences.

Of the various methods of dealing with issues of over plotting, range and scalability, we have been prototyping multi-level tiles with a binning approach combined with a “spiral” ramp [2] for a color ramp. The ramp is perceptually efficient at presenting a sequence and for differentiating the greatest range of values.

2.1 Scatter Plots for Big Data

Scatter plots are an intuitive, easy to use, and widely understood tool for EDA. However, as the data plotted get larger, they suffer from over plotting, as pictured in Figure 1, where true quantities and distributions become obscured.

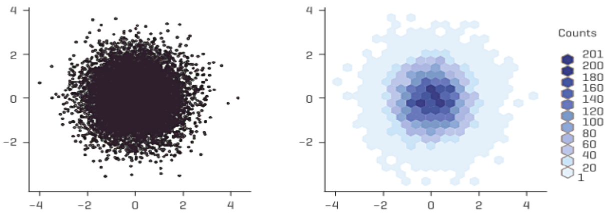

Figure 1: Over plotting (left) addressed with hexbinning (right) [3].

Additionally, including extreme values will often hide important structure, as seen in Figure 2, but excluding them may hide important clues. Local data can be lost at global scales, and while data can be re-plotted at different scales, it is typically not interactive.

Scalability issues also come into play. Plotting tools such as Python’s matplotlib and R are typically limited to datasets that fit

* email: {pschretlen, nkronenfeld, dgray, jmcgeachie, ehall, dcheng, ncovello, bwright} @oculusinfo.com

in memory. Cycle times can be long for large plots, and the loss of interaction hinders exploration.

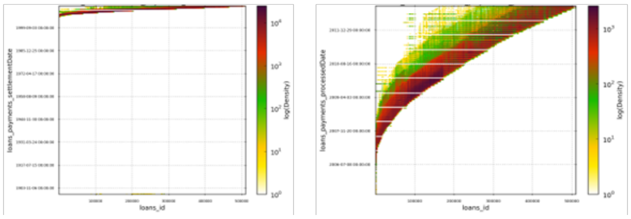

Figure 2: Showing all data (left) with extreme ranges hides structure of interest (right). "Plot Data" results on Kiva data.

2.2 Maps and Scatter Plots

Maps share much in common with scatter plots. Both feature continuous data along two dimensions, use of layering and legends, axes and scales. In fact, scatter plots of geospatial data form maps (Figure 3).

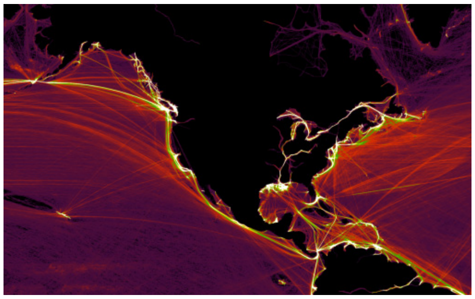

Figure 3: Is this a map or a scatter plot? It is a scatter plot, showing 112M Automatic Identification System (AIS) vessel positions.

2.3 Web Tile Rendering for Big Data Scatter Plots

Solutions for tile rendering used for web friendly maps are portable to scatter plots. Tile generation and distribution that allow for interactively viewing terabytes of satellite imagery can be applied to billions of data points in a scatter plot. Interactive zoom and pan enable navigation and exploration, whether zooming in from country to city streets or into a dense region of a scatter plot to see detail. User experience is optimized with intuitive interactions, controls and responsive views. Client side libraries make it easy to integrate maps into applications, and to add data as raster or vector layers (e.g. OpenLayers, Leaflet.js). Mature tools are available for generating custom base layer tile sets (e.g. TileMill). Alternative representations are supported for base layers, for example satellite vs. road map layers, or in the case of scatter plots, raw plotted points vs. heat maps or kernel smoothing. Vector layers are used to provide points, lines or regions on a map. For a scatter plot they can include models, clusters or annotations.

We use Spark or R to calculate the tiles in a data view model. Count data is put in bins at varying resolution. At the lowest level we are examining every data element. At higher levels we bin and map density to color intensity to indicate quantity of data (Figure 4). Tiles are rendered on demand from the data view.

Figure 4 shows an example of using tile rendering for a scatter plot of 37M Bitcoin transactions (3.6 GB). Tiles are computed at multiple levels of detail. Each level is a new binning of data. Gross higher level features resolve into finer details as you zoom. Using a tiled map as a scatter plot encourages exploration through pan and zoom. The display is effective even though it takes little space.

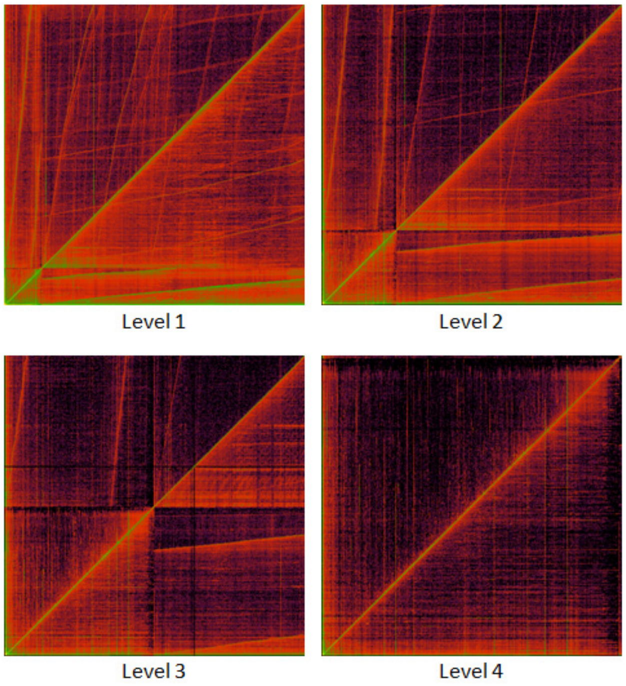

Figure 4: Using a tiled renderer for scatter plots encourages exploration and discovery. "Plot Data" results on Bitcoin data. Source id is plotted against destination id.

For each tile we set 256 by 256 bins. Each level has four times the tiles of the previous level. Data often appears continuous when zoomed out, but becomes discretized at small scale. Only enough levels are needed to reach the point of discretization, since little is gained by zooming in further. For a set of 36M unique data points, for example, level 5 with 1,024 tiles and 67,108,864 bins, is the first level with an average of less than one point per bin.

A high memory cloud configuration is used for computing, with 4 GB/processor, 16 processors/node and 20 nodes. Processing of a 4 GB dataset required 2.23 minutes to complete 42,191 tiles at level 9, and 4.62 minutes for 101,531 tiles at level 10.

3 Conclusion

This work demonstrates interactive "big data" exploration with continuous zooming. Data characterization is made more intuitive through effective visualized aggregation techniques without losing detail. All the data is plotted. The tiled rendering technique is applicable to any size large dataset.

One next step will be to compute feature extractors (e.g. clusterers) and annotate the tiles with additional layers.

Acknowledgement

This study was supported by the Defense Advanced Research Projects Agency (DARPA) under Contract Number FA8750-12-C-0317. The views, opinions, and findings contained in this report are those of the authors and should not be construed as an official Department of Defense position, policy, or decision.

References

[1] Tukey, J, We Need Both Exploratory and Confirmatory. *The American Statistician*, Vol. 34, No. 1, February 1980.

[2] Ware, C., Information Visualization: Perception for Design, Third Edition, Morgan Kaufman, 2013.

[3] Heer, J. and S. Kandel. Interactive Analysis of Big Data, *XRDS*, Vol. 19, No. 1, 2012.