UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

# Exploring Meaningful Scatterplots using Zenvisage

by

Jaewoo Kim

A Senior Thesis submitted for the
Bachelors of Science

in the

DEPARTMENT OF COMPUTER SCIENCE

May 2018

# *Abstract*

DEPARTMENT OF COMPUTER SCIENCE

Bachelors of Science

by Jaewoo Kim

The increasing availability of rich and complex data in a variety of scientific domains poses a pressing need for tools to enable scientists to rapidly make sense of and gather insights from data. One proposed solution is to design visual query systems (VQSs) that allow scientists to search for desired patterns in their data-sets. While many existing VQSs promise to accelerate exploratory data analysis by facilitating this search, they are unfortunately not widely used in practice. We discovered one difficulty of using VQSs stems from the limited support for scatter plot analysis. Our research aims to integrate scatter plot analysis into the Zenvisage system to pave the way for exploration of scatter plots through VQSs.

# Contents

# Chapter 1

# Introduction

One potential solution for the challenge of manually exploring a large collection of visualizations are systems that allow users to specify desired visual patterns, via a high-level specification language or interface, with the system automatically traversing all potential visualization candidates to find and return those that match the specification. We define such systems to be Visual Query Systems, or VQSs for short. One such VQS developed by us is Zenvisage [4], which supports multiple modes of specification, including a sketching canvas where users can draw a pattern of interest, or drag and drop an existing visualization onto the same canvas, with the system finding visualizations that are similar to the queried pattern.

Currently, Zenvisage only provides the above mentioned features for time series plots. Enhancing Zenvisage to support additional types of visualizations such as scatterplots is important because selecting the correct method for plotting data can impact results during analysis. For instance, if there is no reason to believe that one or more attributes of the data-set gives a natural ordering of the data, using time series plots can potentially hide interesting relationships and insights present in the data. Scatterplots are the solution to deal with such data that cannot be represented as a function. That is, mapping every point in a 2D plane reveals various non-linear characteristics of the data. The necessity for scatterplots has been confirmed while working with domain experts to find out possible improvements for Zenvisage. Our Material Science collaborators at CMU suggested that scatterplots are suitable for battery science data exploration since their data is not aggregated along the y axis. The natural step was to study the current state of scatterplot analysis, and the feasibility of its integration to Zenvisage. This paper shares the results of integrating scatterplots into Zenvisage and discusses possible future directions for research.

# Chapter 2

# Related Work

Our work is related to prior literature in scagnostics and human perception of similarity among scatterplots.

## 2.1 Scagnostics as metrics for scatterplot similarity

Exploring large collections of scatterplots have been researched in recent work. They state that as the number of attributes grows for large data sets, the number of scatterplots in the scatterplot matrix (SPLOM) increase in the order of $\mathrm{O}(p^2)$ where $p$ is the number of attributes. An unwieldy number of scatterplots is problematic for two reasons. Visualizing the SPLOM in a fixed number of pixels becomes challenging with large numbers of scatterplots. Also, it will be difficult for data scientists to explore each scatterplot in the SPLOM to find the desired pattern. To address these issues, various metrics called scagnostics [2] were formulated to describe the characteristics of a given scatterplot. The scagnostics include measures for outlying, monotonic, straight, skinny, stringy, striated, clumpy, convex, and skewed scatterplots. The SPLOM of scagnostic measures for each scatterplot will only require $\mathrm{O}(k^2)$ number of scatterplots where $k$ is the number of different scagnostic measures. The exploration time is reduced by comparing numerical summarizations of each visualization rather than the visualizations themselves.

The relevance of scagnostics to Zenvisage is strong because scagnostics can be used as a similarity metric for scatterplots. Similarity of scatterplots can be determined by how similar the scaganostics measures are for two given plots. In Zenvisage, similarity metrics such as Euclidean distance are already used for comparing time series plots. Substitution of scagnostics for existing similarity metrics allows use of analysis procedures and optimizations which are already implemented in Zenvisage.

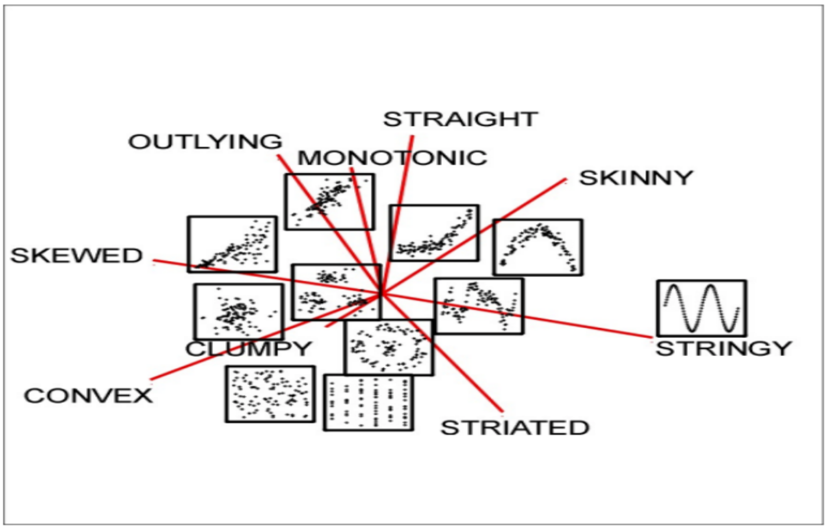

Figure 2.1: Bi plot of different scagnostics

## 2.2 Human perception of similarity among scatterplots

Recent studies evaluate to what extent scagnostics match human perception of similarities between scatterplots [1]. The studies conduct a similarity perception study involving 18 participants with scientific backgrounds. Results show that perceived similarity and scagnostics based similarity have a weak correlation. The most common metrics for perceived similarities were summarized by density, orientation, spread, regularity, groupings, edges. Discrepancy between human perceived similarity and scagnostics based similarity implies that ranking scatterplots with certain scagnostics may lead to less intuitive results.

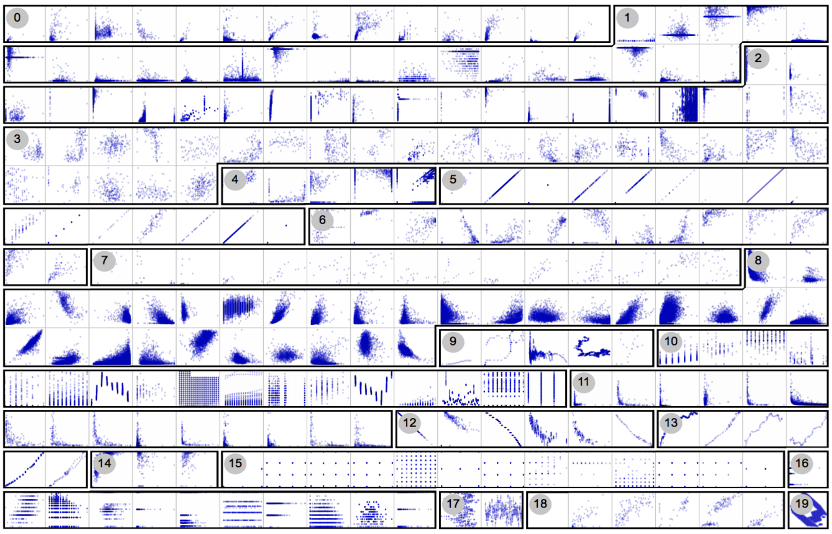

Figure 2.2: scatterplots grouped by percieved similarity

Chapter 3

# Development Process

## 3.1 Bounding Polygon Query

Our main goal for the current iteration of integrating scatterplots is to build the query work-flow rather than to implement complex queries. We kept the overhead for queries to a minimum by implementing a simple query. Inspired by how users draw queries into the sketchpad for series plots in Zenvisage, we implemented a bounding polygon query. A demo of the query is provided in a later section. Users are able to draw arbitrary polygons on top of scatterplots to submit a query. The system processes the query by counting the number of points within the bounding polygon for each plot. The plots are ranked based on scores computed with these counts.

The ideal design should fit into the general work-flow of Zenvisage, while providing extensibility in the types of queries it can support. Both front-end and back-end functionality address these goals. Providing the sketchpad interface for scatterplot queries maintains a uniform interface pattern by mirroring series plot queries and modular design in the back-end allows addition of new ranking functions.

## 3.2 Front-end Development

The front-end development consisted of the following components

- **Binning**: To visualize large data sets, a scatterplot is divided into equal sized hexagonal bins. The count of data points which lay within each hexagon determines how the hexagon will be colored. A linear scale is used to map counts of data points to a color hue. Higher saturation means higher counts of data points within a hexagon.

- **Polygon Drawing**: The user can draw any arbitrary polygon within the sketchpad by clicking multiple points. The user double clicks the sketchpad to finish drawing a polygon. A bounding polygon query with input as the sketched polygon is triggered by the double-click.

- **Query Submission**: The front-end must send a new type of query to the back-end. The new query must request for scatterplots instead of series plots and include input such as polygon points and ranking functions.

## 3.3 Back-end Development

- **Process Query**: The query from the front-end is processed for its inputs such as visualization method, polygon points, and ranking functions. The system create scatter nodes for its execution graph only if the visualization method variable is set to "scatter". Polygon points are saved for use in bounding polygon queries. Ranking functions determine which type of ranking will be used to sort the output results.

- **Retrieving candidate scatterplots using ZQL [4]**: The front-end query is also used to submit a query to the ZQL engine to retrieve candidate scatterplots. The ZQL engine returns scatterplots with varying z axis.

- **Ranking function**: Once candidate scatterplots are retrieved, the plots are sorted in a ranking function. Currently, the ranking function ranks plots with the most data points within a bounding polygon as the highest rank. Other ranking functions can be substituted, and the complexity ranking algorithm impact running time of processing queries.

# Chapter 4

# Results

## 4.1 Demo of bounding box

The following figures present how one may use the bounding polygon query. For example, a user may want to know which class of batteries have the most data points in the bottom left corner of the scatterplots. The corresponding bounding polygon query can be drawn as in Figure 4.1.

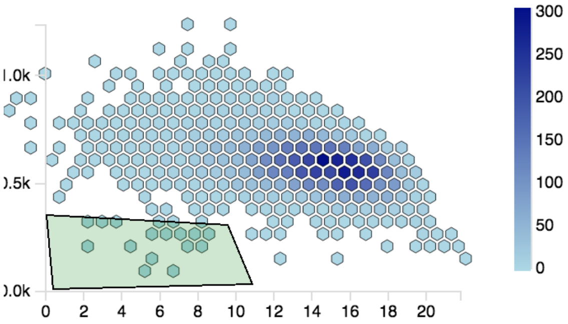

FIGURE 4.1: Bounding polygon query showing hexbins for data points and green polygon for bounding polygon. Colorbar shows the value of colors for hexbins.

The results returned by Zenvisage are shown in Figure 4.2. By inspection, it is visible that scatterplots with more data points on the bottom left corner are highly ranked than plots with missing data points in the region.

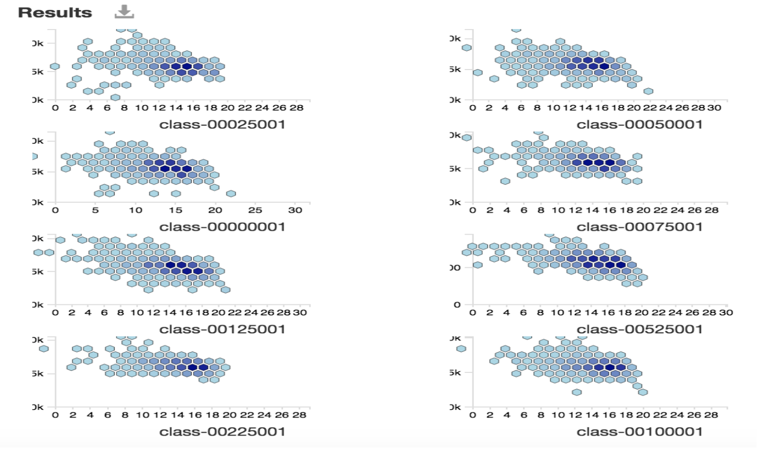

FIGURE 4.2: Bounding polygon query results

## 4.2 Verification of results

To validate results from the bounding polygon query instead of relying on inspection, an synthetic toy data set was used for testing purposes. The data set only has three z axes with one data point for each z value. Bounding one point as shown in Figure 4.3 with a polygon should rank the scatterplot which contains the bounded point to be first.

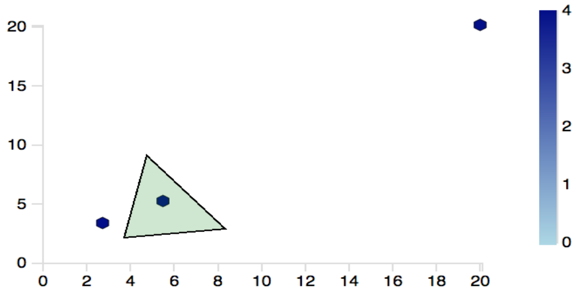

FIGURE 4.3: Test query submission for verification

Results returned are shown in Figure 4.4. The results indicate that our algorithm is working correctly since the scatterplot which contains data point (5,6) is ranked first.

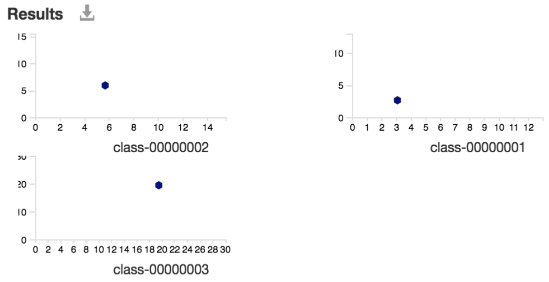

FIGURE 4.4: Test query results

# Chapter 5

# Discussion and Future Work

## 5.1 Addition of Scagnostics Queries

Currently, types of queries that can be submitted by the user is limited to bounding polygon queries. Extending ranking functions to support scagnostics will increase the number of patterns that users can search for. For instance, a user who wishes to search monotonic scatterplots can select "monotonic" as the metric for similarity search, and the results of the query will rank monotonic scatterplots first.

Implementing scagnostics will reveal scalability issues that exist in current work-flow. Since bounding polygon queries only run in O($n$) for counting data points inside the polygon, and O($nlogn$) for ranking different procedures, testing our system with scagnostics can reveal possible bottlenecks in the work-flow which have to be optimized.

## 5.2 User-studies

While different methods of defining scatterplot similarity have been researched, similarity search of scatterplots in the context of interactive VQSs have not been studied. User studies with real-world analysts who heavily use scatterplots may lead to formulations of new similarity metrics and improvements in the interface.

## 5.3 New Front-end features

Many front-end features still need to be implemented. One feature is the addition of color bars to the query results. However, having color bars for every result may clutter the results pane and affect the user's experience. Two possible solutions are to increase the size of the results pane for scatterplot queries and adding zooming functions for each plot. More real estate in the results pane may alleviate the problem of plots and color bars being cluttered. Adding zooming functions allows users to explore each scatterplot in detail if the initial plots are too small.

Another possible addition to front-end features is a slider which allows control of binning properties. Depending on the input value from the slider, the user can set the size of hexagons for binning data points. As a result, users can specify the level of granularity for aggregating data-points in scatterplots. Bin color is another property that the user may change using a drop down menu. Users should be able to change colors that represent binning values if certain hues are difficult to perceive.

# Bibliography

[1] Anshul Vikram Pandey, Josua Krause, Cristian Felix, Jeremy Boy and Enrico Bertini. 1993. *Towards Understanding Human Similarity Perception in the Analysis of Large Sets of Scatter Plots.* CHI 16

[2] Leland Wilkinson, Anushka Anand and Robert Grossman. 2005. *Graph-Theoretic Scagnostics.* InfoVis 05

[3] Doris Jung-Lin Lee, John Lee, Tarique Siddiqui, Jaewoo Kim, Karrie Karahalios, Aditya Parameswaran. 2017. *Accelerating Scientific Data Exploration via Visual Query System.* TVCG

[4] Tarique Siddiqui, Albert Kim, John Lee, Karrie Karahalios, and Aditya Parameswaran. 2016. *Effortless data exploration with zenvisage: an expressive and interactive visual analytics system.* Proceedings of the VLDB Endowment 10