CSCI 6406 Visualization – Project Progress Report
Jéssica Pauli de C. Bonson

Introduction

The goal of this project is to develop a visualization tool for the analysis of classification data for machine learning problems. The tool will have a scatter matrix, histograms in the diagonal of the matrix, and a star plot. The histograms show the data distribution per attribute and the star plot shows the attribute values for the selected data points. The sketch of the tool is shown on Figure 1, a bigger version can be found in the project proposal.

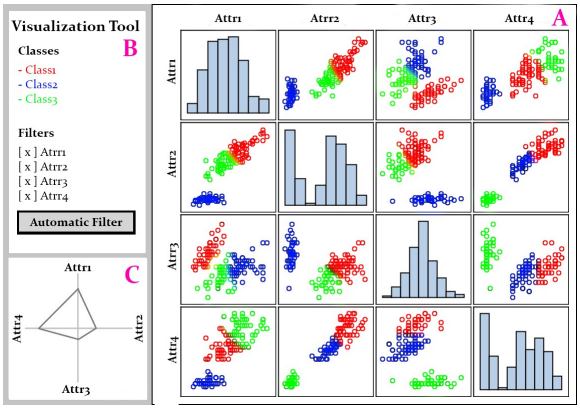

Figure 1. Mockup of the system

The system will have three features: selection, filtering, and zooming. The selection feature enables the user to select points in a chart of the scatter matrix, the same points in the other charts will also be selected, and the star plot will be modified to represent the current selection. The filtering changes the visualization in the charts according to what attributes interest the user, and can be manual (by selecting the attributes to be shown) or automatic (an algorithm automatically filters the attributes it believes to be irrelevant). The zooming will allow the user to occupy all the matrix space with only the chart of interest.
The tool is being developed using Python and Javascript. I will use the web framework Flask and the Javascript library D3.

Current progress

Figure 2 shows the current state of the system:

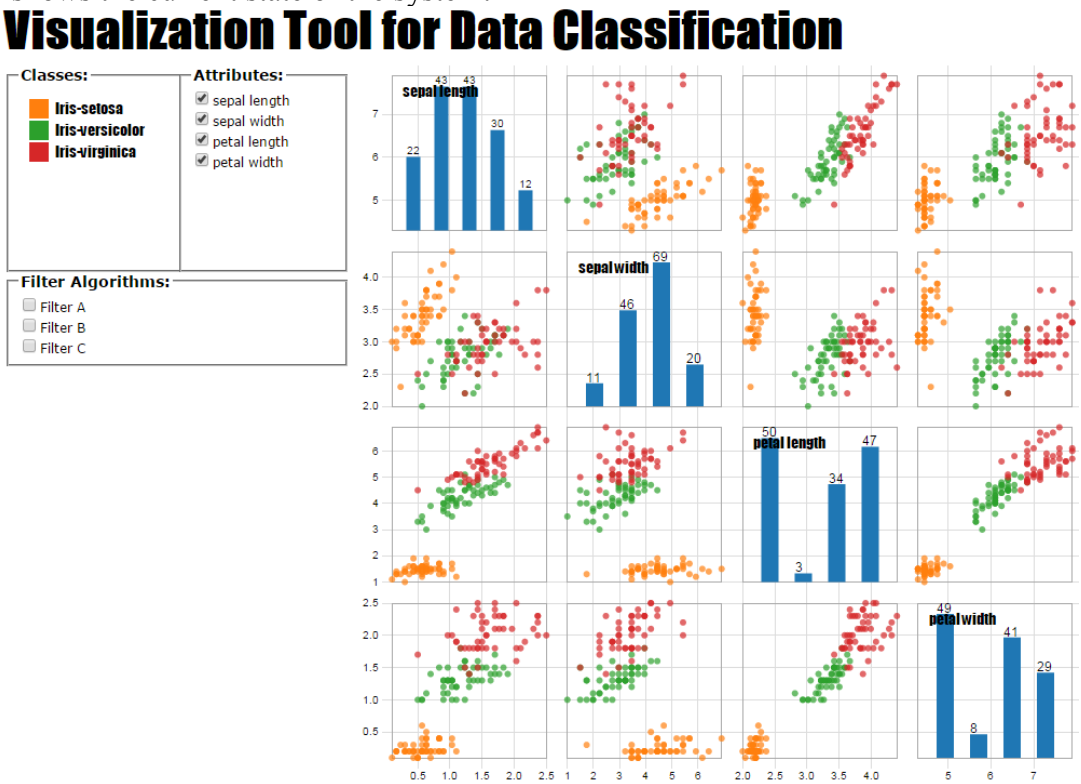

Figure 2. Visualization system for Iris dataset

The system will be exemplified using the Iris dataset, but it is able to run with any real-valued classification dataset in CSV. For example, Figure 3 shows the systems loaded with the Ecoli dataset.

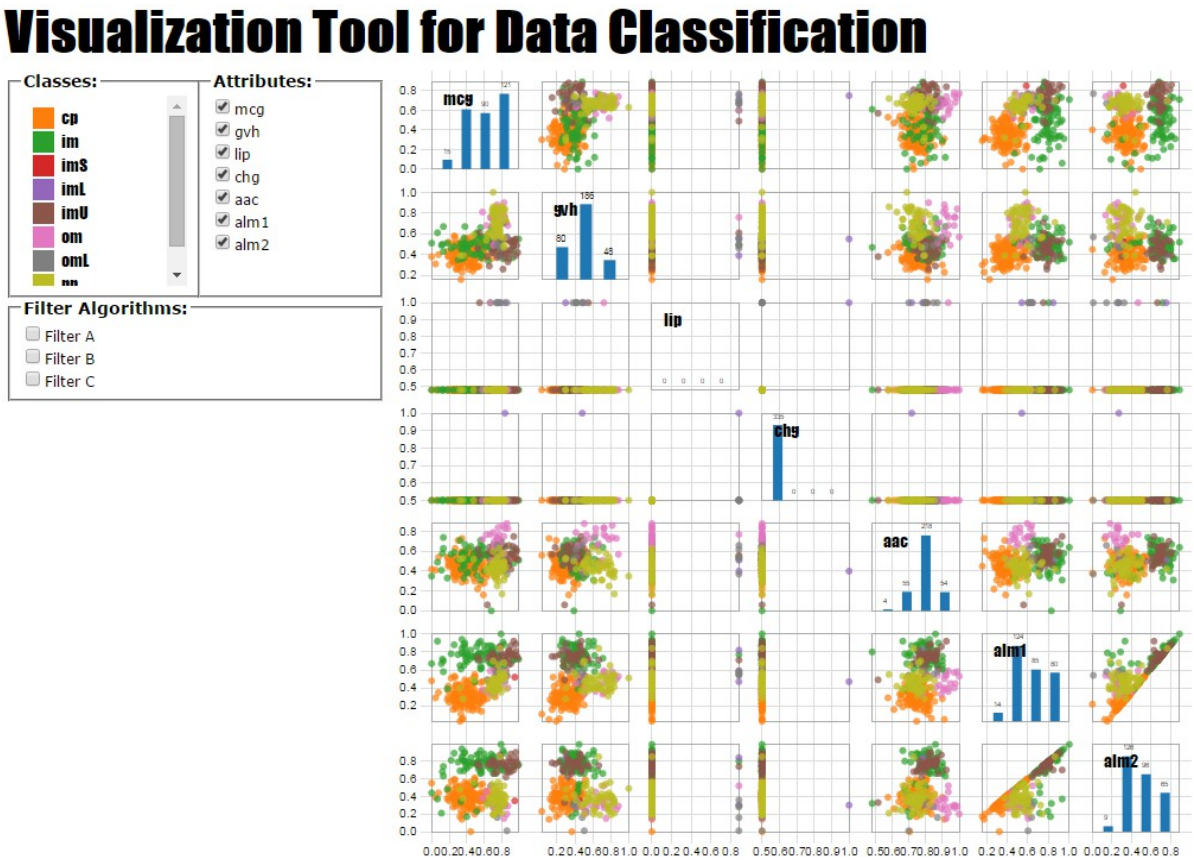

Figure 3 Visualization system for Ecoli dataset

The data representation using scatter matrix and histograms is implemented. Everything is done regarding the scatter plots, but there are still some fine-tuning that is necessary to be made regarding the histograms. Figure 4 shows the histogram for the attribute 'sepal length'. The data distribution is calculated for five bars, each bar has a tooltip that shows the range of values is that bar, and has the total of data points inside that range on top of the bar. The font size dynamically changes to fit the bar width. The last thing to fine-tune is that for bigger zooms the font stops being on top of the bar, as shown in the Figure 5. This figure was reduced to better fit the report, but it occupies all the scatter matrix space.

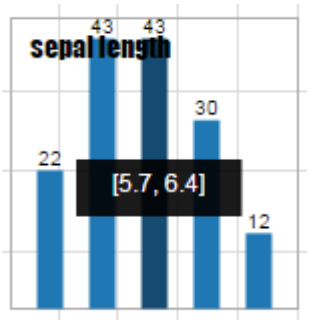

Figure 4. Selection feature for histograms

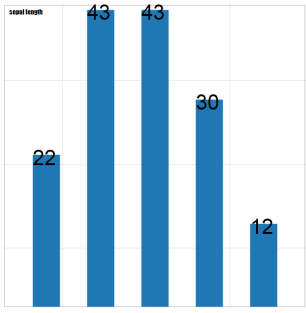

Figure 5. Problem with great zooms

The selection feature and the manual filtering are finished, but only affect the scatter matrix and the histograms since the star plot isn't implemented yet. Figure 6 shows the selection of data points in the scatter matrix. All the points representing the same data are highlighted across the charts while the other points are hidden.

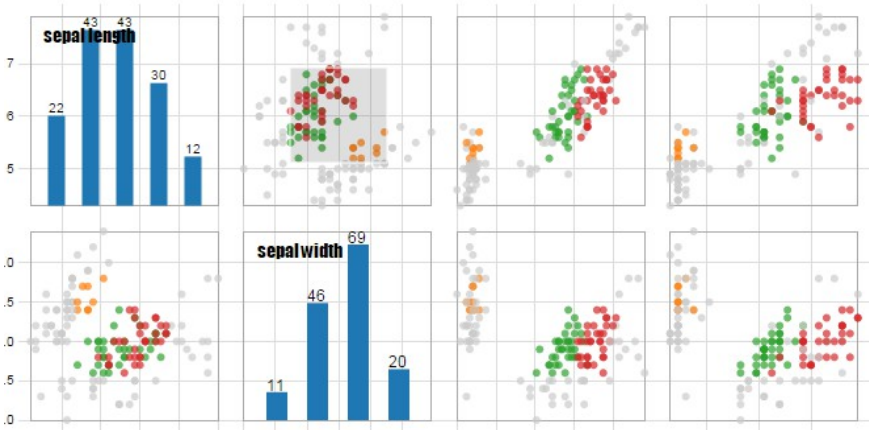

Figure 6. Selection feature for scatter plots

Figure 7 demonstrates the manual filtering of two attributes (sepal width and petal length) by unchecking the respective checkboxes, so the other two attributes (sepal length and petal width) fit all the scatter matrix space.

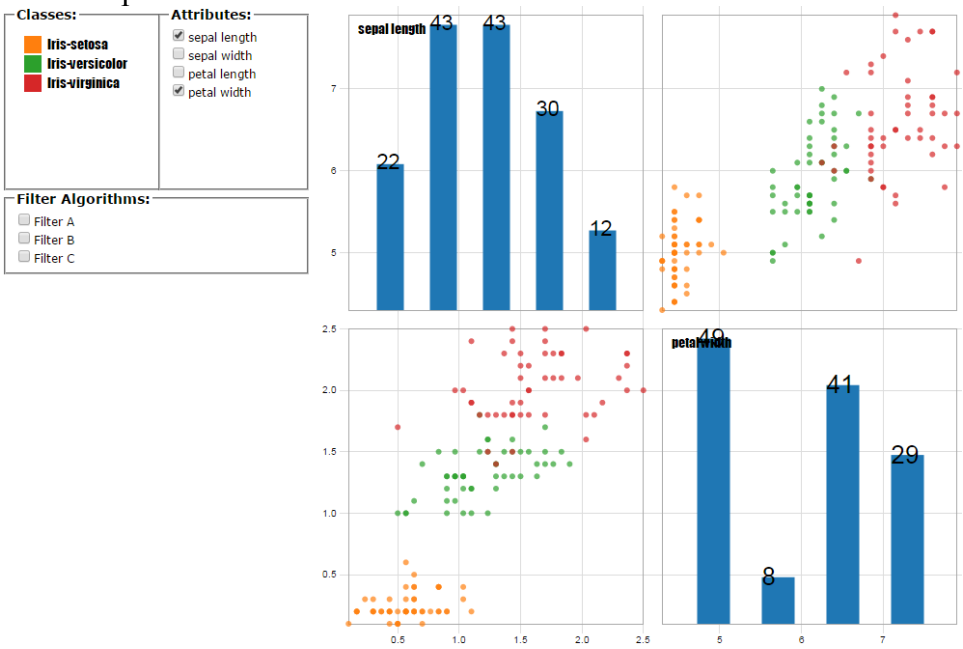

Figure 7. Filtered two attributes from the Iris dataset.

Future Work

The main implementation that is missing is the star plot, along with it being affected by the selection and filter features. It seems that d3 doesn't offer anything that eases the work of implementing a star plot, so it will be more complicated than the scatter matrix and the histograms. Another feature still to be implemented is the zooming, along with figuring out how the user will activate and deactivate it, since clicking on the scatter plots already activates the selection feature. Lastly, it is necessary to implement the automatic filtering using machine learning algorithms. To do this, it will be necessary to integrate the algorithms on the server side and use Ajax to call them when the user chooses one of the automatic filtering methods. The UI for “Filter Algorithms” probably will also be changed to better integrate them.

Main Task List Summary:

- implement the star plot
- star plot working with selection
- star plot working with filtering
- implement zooming + UI adaptation
- implement automatic filtering + UI adaptation
- fine-tune the histograms for greater zooms

In case there is enough time, there are other tasks that I will implement to improve the tool.

Optional Task List Summary:

- option to choose the size of the scatter matrix, so it can be better visualized for datasets with lots of features
- option to upload .csv files with datasets to the system
- filtering feature for classes, so the data for some classes could be hidden to remove noise
- dynamic histograms that change the quantity of bars depending on the zoom
- updated the histograms to use stacked bars, so each bar will have the quantity per range and per class
- automatic feature extraction