Submission - Exercise [6]

Visual Data Analysis

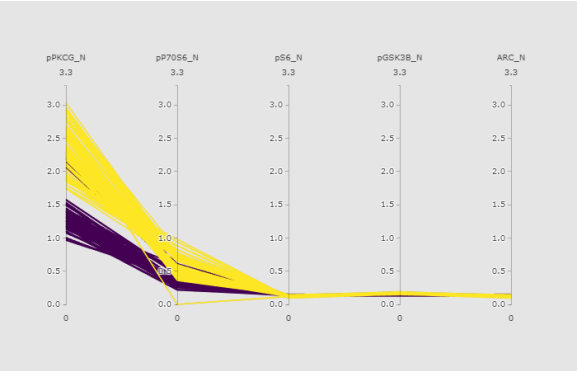
[Maryam Assaedi, maryam.assaedi@rwth-aachen.de]
[Mst. Mahfuja Akter , s6msakte@uni-bonn.de]
[Mahpara Hyder Chowdhury, s6machow@uni-bonn.de]

May 12, 2019

Exercise 1(Parallel Coordinates Plot in Plotly)

a) Read the file Data_Cortex_Nuclear.xls that we previously used in Exercise 1b) of Sheet 5. Extract subgroups t-CS-s and c-CS-s. Use plotly to create a parallel coordinates plot from the following 5 proteins: (pPKCG_N, pP70S6_N, pS6, pGSK3B_N, ARC_N). Assign different colors to the two selected classes. Annotate every axis with the correct protein name.

We have used Data_Cortex_Nuclear dataset for this task. At first, we have added 0 values for null values. Then, we have subgrouped to t-CS-s and c-CS-s. After that, using plotly package, we have created a Parallel Coordinates plot for the five following proteins: (pPKCG_N, pP70S6_N, pS6, pGSK3B_N, ARC_N). Lastly, we have added different colors to the two particular classes and added axis names to the protein names accordingly.



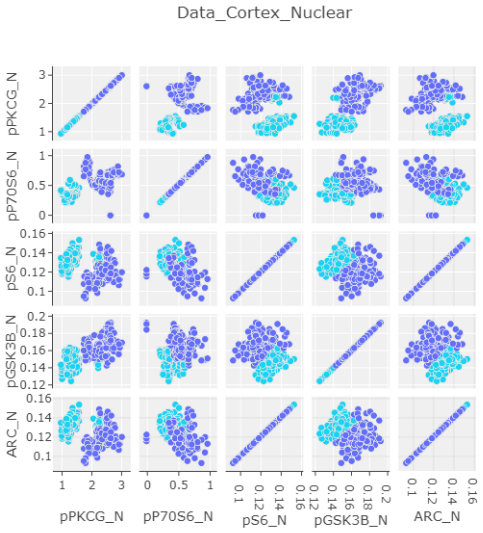
b) Explore the data by interacting with the parallel coordinates plot. Do you find anything suspicious about the data set?

Yes, there is one data point which seems separated from the rest of the samples that we can see in pP70S6_N protein.

Exercise 2 (Scatterplot Matrix in Dash)

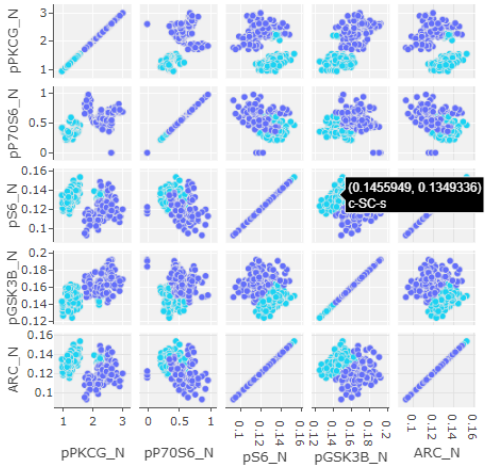
a) Use the same dataset, subgroups, and proteins as in Exercise 1. Use the dash framework to create a $5 * 5$ scatterplot matrix, for which the diagonal plots represent the histogram of the corresponding attribute. Assign a separate color to each class and annotate the axis.

A $5 * 5$ scatterplot matrix has been created for the five pro tines pPKCG_N, pP70S6_N, pS6, pGSK3B_N, ARC_N with subgroups from the Data_Cortex_Nuclear dataset. The two classes has been separated using two different colors and the axis are named accordingly.



b) In the non diagonal cells, visualize the scatterplots of corresponding pair. Assign corresponding name and text to each sample on scatterplot, so that the exact values of a point and its class are shown when moving the mouse over it.

In the non-diagonal cells, we can see the values of each data point with corresponding class name by hovering the mouse over it. Below is a screen-shot given.



Exercise 3 (Large-Scale Graph Visualization)

a) The ZAME visualization tool uses a specific hierarchical data structure for storing graphs at multiple scales. At the lowest level, four integers are stored per vertex, and either six or four per edge. What do these integers describe? Which two are optional in case of the edges, and what is their purpose?

In the ZAME architecture, there exists two tables, one table to save the vertices and the other one stores the edges. The vertex table stores 4 integers per vertex describing the first and last edge for the two linked edge lists for incoming and outgoing edges.

The edges table contains 4 integers to store the source and sink vertices for directed graphs. And two extra integers which can be omitted in order to save memory and they describe a back-link.

b) The zoomable edge table stores edges in a particular order that makes it fast to search for an edge given its vertices. Write efficient pseudocode that returns the index within this table of an edge connecting vertices u and v, and returns None if the table does not contain such an edge.

For this task we have a main function FindEdge that takes in the two vertices and loops on all the levels. Each time it calls the iterate function which gives all the element in that given level. Then it calls the BinarySearch function and passing the inputs u, v and iterator. The BinarySearch function outputs a boolean variable "found" and an edge. If the edge is found, the FindEdge returns the edge, else it returns None.

Algorithm 1 FindEdge(u,v)

```
for i=log_2(||V||) to 0 do
    iterator = iterator(i)
    found,edge = BinarySearch(u,v,iterator)
    if found == True then
        return edge
    else
        return None
    end
end
```

The BinarySearch function takes in the u,v and iterator. It uses binary search on the iterator to find the edges that have u as their source vertex. When such edges are found, another binary search is used on all edges (with source = u) to find the edge that has v as its sink vertex. When said edge is found, the function returns true and the edge. If no edge is found then it returns false and None.

Algorithm 2 BinarySearch(u,v,iterator)

```
for i=log2(||V||)to 0 do
    first = 0
    last = iterator.size-1
    while first ≤ last do
        middle = (first+last)/2
        if iterator[middle].sourceVertex¿u then
            last = middle -1
        else
            if iterator[middle].sourceVertex¡u then
                last=middle+1
            else
                firstPositionU=middle
                break
            end
        end
    end
    lastPositionU=middle
    while iterator[lastPositionU.sourceVertex == u do
        lastPositionU++
    end
    lastPositionU -=1
    while firstPositionU ≤ lastPositionU do
        middle = (firstPositionU+ lastPositionU)/2
        if iterator[middle].sinkVertex¿v then
            lastPositionU=middle-1
        else
            if iterator[middle].sinkVertex¡v then
                lastPositionU=middle+1
            else
                return (True,iterator[middle])
            end
        end
    end
    return (False,None)
end
```

c) In the pseudocode listed in the paper's Figure 4, some modifications are highlighted in boldface, on lines starting with a bar. What is the purpose of these modifications?

The modified version of PCA by Harel and Koren had a problem with the way choosing the pivots. It was in some cases inefficient as it only chooses the farthest leaf from the pivot. To avoid this problem, the authors of the paper decided to penalize the edges according to the number of times they participate the path between existing pivots. This is done by halving the edges that are in the shortest path selected for each pivot pair. By doing so, they hope that the algorithm will explore other leafs and have a good representation of the overall structure.

d) Why is the Traveling Salesman Problem relevant to adjacency matrix based graph visualization?

In the Traveling Salesman Problem, we want to find the shortest distance between several nodes so we can traverse the whole set of points without crossing previous paths. This is similar to adjacency matrix based graph visualization in the sense that we also want to find the closest nodes so we can be able to reorder them.

e) How does ZAME aggregate nominal attributes? Why is that problematic?

Nominal data is data that does not represent any pattern between different entities. In ZAME, nominal data is aggregated by selecting first label to represent the whole aggregation. This is problematic as data is aggregated based on only the first label which does not say anything about how close the data is.

f) What is the role of tile management within the ZAME system? What is an LRU cache?

Storing all the nodes in texture memory is impossible and thus the full matrix is split into tiles of fixed size. This provides a way to efficiently load the texture for individual tiles depending on user navigation.

The LRU cache is responsible for tracking which tiles are in use and which can be recycled. If the user would like to view a previously cached tile, then this tile is fetched from memory and displayed without further costs.

g) What is the difference between geometric zoom and detail zoom in the system?

Geometric Zoom defines the viewport which states which part of the matrix is mapped to the screen.

Detail Zoom states which details is viewed according to which hierarchical level we are zooming to.

Exercise 4(Visualizing Trees and Graphs)

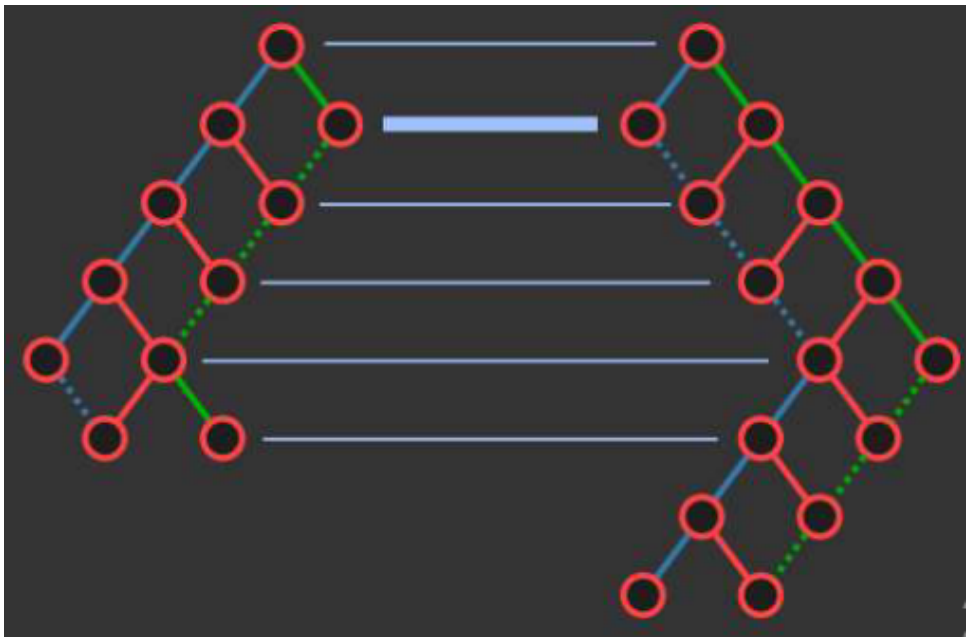
a) Force-directed graph layout uses an electrical repulsion term that is often set to zero for nodes whose current distance exceeds a certain threshold. Name two reasons for this truncation.

Two truncation reason of certain threshold:

1. To avoid high distance calculation between nodes and thus reducing the graph generation time.
2. Impose a maximum distance by isolating the node which keeps nodes within specific image size.

b) What is a thread in the context of the Reingold-Tilford algorithm? What purpose does it serve?

In context of Reingold-Tilford algorithm, each leaf of the tree that has a successor in the same contour, thread is the pointer of this successor. In figure, threads are represented by dotted arrows. In this process, for every nodes of a contour, we have a pointer of it's successor: either it is the leftmost(rightmost) child or it is given by thread. Finally add a new thread whenever two sub trees of different height are combined. Thus it reduces the traversing time of contours by tracking their successor by pointer.



c) Briefly explain a benefit and a drawback of the squarified treemaps algorithm compared to classical treemaps.
Benefits: ST sorts the sibling node by their size and generated environments are realistic which is very useful for virtual humans simulation.
Drawbacks: For larger layered treemaps (e.g File system), the hierarchical structure is more difficult to visualize.

Exercise 5(Marks and Channels)

a) Name two visualization channels that are expressive for ordered attributes, and rank them according to their effectiveness. Name another channel that is expressive for categorical attributes.
Two most effective visualization channel for ordered attributes are:
1. Position on common scale.
2. Length (1D) size.

One expressive channel for categorical attribute is:
1. Spatial region.

b) Which marks and channels are used in the plot below? How would you modify the plot if viewers were primarily interested in comparing the amount of fiction read by each person?
It's used 'Position on unaligned scale' and color hue in this plot.
If we want to show the comparison of the amount of Fiction reader and Non-fiction reader, we can use 'Position on common scale' and 'Color' hue as well.