Week (9) - “Palettailor: Discriminable Colorization for Categorical Data”

Khulood Alkhudaidi

This paper aims to find a way to automatically generate and specify color palette to categorical visualizations, such as scatterplot, bar chart, and line chart, taking into account the characteristics of the data. In general, finding a discriminable palette is an arduous process where there are many color palette options for the user to choose from. Additionally, even when the user succeeds in finding a discriminable palette, the colors might not be appealing to the readers.

There are many methods for color palette generation; however, these methods do not take the data into account. This paper's researchers created Palettailor, "*a data-aware color palette generation framework, that automatically generates categorical palettes with maximized discriminability for different visualization types*" [1]. They used three functions: Point Distances obtained from the visualization, Name Difference, and Color Discrimination obtained from the palette to achieve their goal.

Method:

Point Distances:

They found that one of the current separation methods uses the K-nearest graph to measure point distinctness and find the optimal solution; however, they noticed that this method works with some visualizations only. So they introduced α-shape graph with a customized simulated annealing algorithm to find an optimal palette for class discrimination in less time.

Name Difference:

The researchers faced the problem of two colors having the same name but perceived differently. For example, "Violet" and "Dark purple" are different perceptually, but both can be called by one name, "Purple." To solve this issue, they used cosine distance to measure the similarity.

Color Discremenation:

Perceiving color differences is impacted by the spatial distance, so they adjusted the optimization, making the color distinguishable even when the distance is very large.

Evaluation:

They evaluated their tool by conducting two experiments to test the discrimination for scatterplot and line chart. They also conducted one experiment to test the users' palette color preferences. All of the experiments were against the existing methods such as *Tableau* and *Colorization*. For the discrimination, Palettailor was the best compared with these tools.

I like what Palettailor offers for the user. When it comes to class discrimination, It saves effort and time with high-quality results. However, I think the color preferences part needs some improvement to make the color palette more appealing. It would be a good addition if Palettailor has a feature where the user can load a palette and get feedback and suggestions for better palette options. I

know Palettailor is a palette *generator*, but I think loading a palette and improving it considering its colors via Palettailor can solve the class discrimination part and, at the same time, give the user appealing options that match their preferences.

Sources:

1. https://arxiv.org/pdf/2009.02969.pdf