Discover Correlation with Parallel Color Bars

I propose a general visualization technique targeted at showing correlations in high dimensional data. It's similar to Parallel Coordinates, but instead of using edges, it uses colors to indicate the connections between values.

Ivan Zhao, ivan.h.zhao@gmail.com

I. Domain, Personal Expertise

I have been working on the Boeing Visual Analytics project at UBC and SFU for over a year, primarily using various off-the-shelf visualization tools (such as Tableau, Inspire, Starlight, Jigsaw) to analyze aviation related datasets.

One central theme of the study was the discovering correlations, which usually comes in two flavors: the first flavor is to *validate the known correlations*. For aircraft engineers, the probabilistic correlation of a component X failure leading to phenomenon Y is usually already known at the design stage of the aircraft (as aircraft failures are unavoidable, but we can minimize their probability to 10^-7 or less through design tradeoffs, which is essentially zero). Therefore, to ensure the engineers have made the correct design tradeoffs, we need to use real world evidence, such as aviation accident/incident data, or airline maintenance data to validate the design hypothesis. In this case, the task is basically a *top-down search* for correlations, in the sense that we know the landing gear on Boeing 737 has a 10^-9 likelihood to fail (the number is made up), but can the empirical evidence we collected correlate to this probability?

The second flavor is to *discover the outliner correlations*. Although flying airplanes are fairly controlled environments, there will always be real world noise that have not been taken accounted by the aircraft designers. Their likelihood to happen might be very small, which makes them very difficult to be detected from maintenance feedbacks, yet the consequence is often unmeasurable.

From my personal experience using pre-existing visualization tools analyzing aviation datasets, Tableau might be the most matured one by far; but I found it difficult to analyze more than 4-5 dimensions in the same visual display (what it calls a *worksheet* in the software), and other tools performed even less adequately.

From this, I suspect there's room to improve on top of current visualization tools, *not* just for analyzing aviation data, but for any high dimensional data in general, especially if the focus is on displaying the correlations. In this project, I am proposing a new (or incrementally new) visualization named *Parallel Color Bar* to address these issues mentioned above.

II. Dataset

Real world data from *FAA (Federal Aviation Administration)*, *North America Birdstrike Dataset*, and other aviation data sources. The desired targeted testing size would contain around 5000 documents (rows), for 20+ dimensions (columns). Most of the datasets contain numerical, nominal and text fields.

III. Task

a. Displaying multiple dimensions; showing the frequency of values in each dimension.
b. Quickly overviewing the entire dataset.
c. Quickly identifying the correlation between values from different dimensions (central theme of this project).
d. Hide uninterested data; and focus on the interesting part.
e. Detecting outliners.

IV. Previous Work

Parallel Coordinates [1, Inselberg] is one of the standard for high dimensional visualization, in which edges are used to indicate the connections between different values in the neighboring dimension axes. However, there are many limitations.

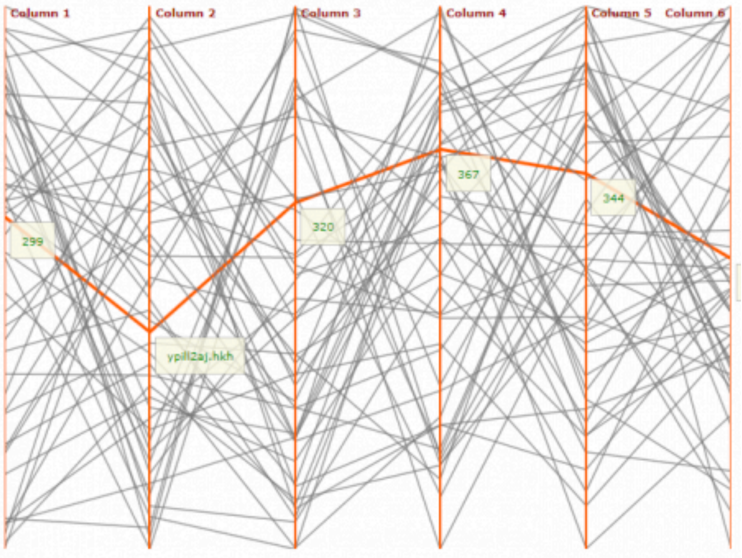
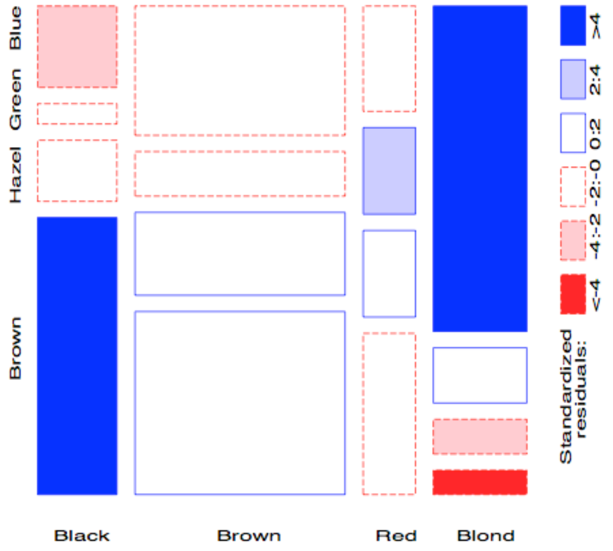


Fig. 1 Parallel Coordinates of 6 dimensions

Parallel Coordinates, Cons:
- Edges are easily to be over-clustered, even with technique such as Edge Bundling [2, Holten]
- Ordering of the dimensions matters.
- Only connections between neighboring dimensions can be shown
- Difficult to represent the frequency of each value (color might help in a limited way)

Besides Parallel Coordinates, structurally this project most resembles the **Mosaic Display** [3]. Below is a Mosaic showing the connections between hair color values (x-direction) and eye color values (y-direction). The width and height of each block is proportional the frequency of the underlying value, hence the area represents the amount of population that satisfying both values.

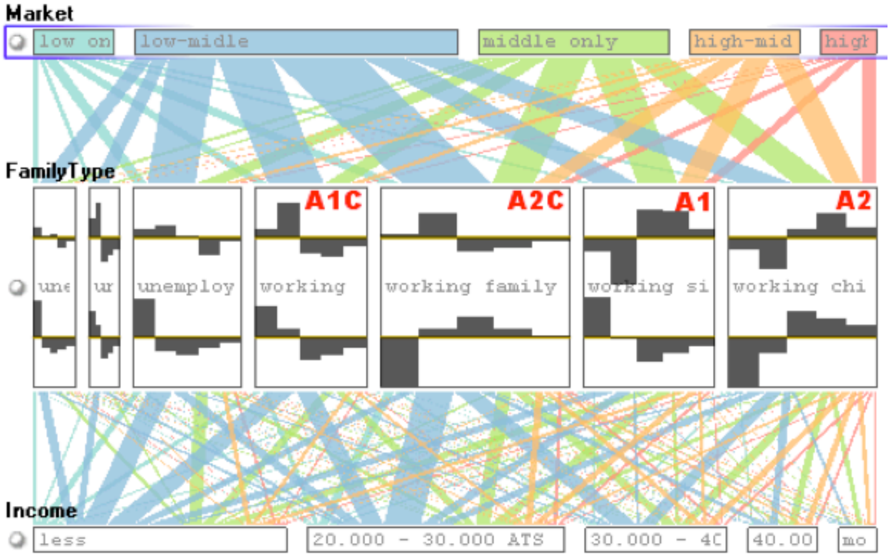
Note: in the figure below, both color and line style indicate the standardized residuals *redundantly*, just in case of b&w printing.



Mosaic Display, Cons:
- Limited to categorical data
- Limited to two dimensions
- Area is difficult to perceive accurately
- Rigid layout; position of the blocks matters
- Lots of space are "wasted" within each block. (It is used to display the standardized residuals in the example, but it might be less interesting to non-statistical users.)

Fig. 2 Mosaic showing the hair color dimension (x-direction), with the eye color dimension.

Last, this project's emphasis on color usage will also resemble the **Parallel Set** [4, Bendix], in which each dimension is laid out horizontally on top of each other, and the values within that dimension is displayed vertically next to each other, with width corespondent to the frequency of that value. The user would indicate a "interested dimension", which will be color-highlighted. Colored edges would then coming from the "interested dimension" to the rest of the graph, with edge width correspondent to the underlying frequency.



Parallel Set, Cons:
- Limited to categorical data
- Messy edges still exist
- Difficult to display more than 3 dimensions

Fig. 3. Parallel Set, displaying the type of household is shown in relation with the income and the user-defined favorite supermarket dimension

V. Discussion

When it comes to displaying correlations, Scatter Plot is the default; but its orthogonal Cartesian axes is expensive as it takes a lot of room for only two dimensions.

Parallel Coordinate in some ways solves this problem by turning the dimensions parallel to each other, saving the room to pack in more dimensions. But as a tradeoff, the information itself has to take up more space (from a *dot* in the Scatter Plot into a *edge)*, which can easily cluster the display.

If we step back and reconsider the issue, in essence, both displays are trying to *connect* dimension A to dimension B (through the logical AND expression), and our information is residing in this *connection* (either as a dot or a edge).

But given our versatile visual perception, why don't we express this connection in some visual attributes other than *displacement* that is used in Scatter Plot and Parallel Coordinate? To name a few, our alternatives are *color, motion, area, texture, order* etc. [5, Cleveland]

The last two examples listed in the IV. Previous Work section are already using some forms of these two visual attributes to express the connections, although with some limitations. For example, the Mosaic Display uses *area* to convey the frequency of the connections; but its orthogonal axes limits it to only two dimensional, categorical data. Parallel Set uses *color* attribute to differentiate the connections, in relation to an "interested dimension" selected by the user; but this coloring process only applies on top of the pre-existing edges, which is not only redundant, but in my opinion fails to reach the full potential that a color oriented design alone can achieve.

VI. Proposed Solution & Illustration

Studies have shown that if color of the targeted stimulus was known to the user, it improves both the accuracy and speed of the visual search, and possibly more effective than other achromatic attributes such as brightness, size, shape etc [6, Christ]. In this project, I am proposing a design using *color attribute* to present the connections between dimensions. An illustration of the prototype is shown below.

Let's first focus on Fig. 4. Similar to Parallel Coordinates, dimensions are laid out vertically next to each other, all with the same width. The categorical values in each dimension are presented by blocks, stacked on top of each other, with heights proportional to their frequencies.

To display numerical values, it would use user defined ranges to bin the data into categories and display them in the same fashion, however this detail is not yet clear at this stage of this project.

Let's now move to Fig.5. Similar to Parallel Set, the user can select one dimension as the coloring reference (Dim 4). Then, the blocks in all other dimensions will be vertically divided into many sub-blocks, colored based on the interested dimension, and each with a width proportional to the frequency that satisfy the values in both dimensions.

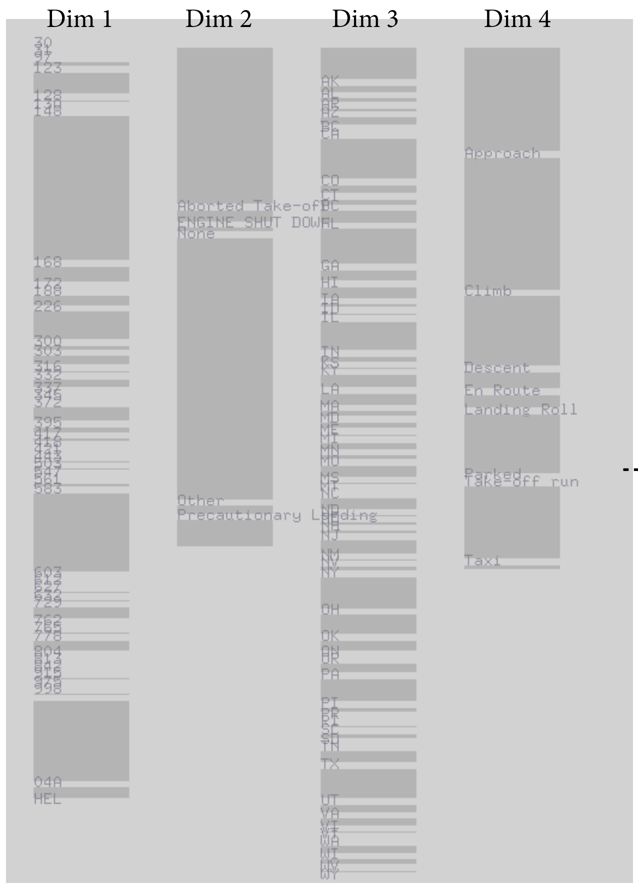


Fig. 4. Parallel Color Bar with 4 dimensions, before focusing on any dimension as the color reference.

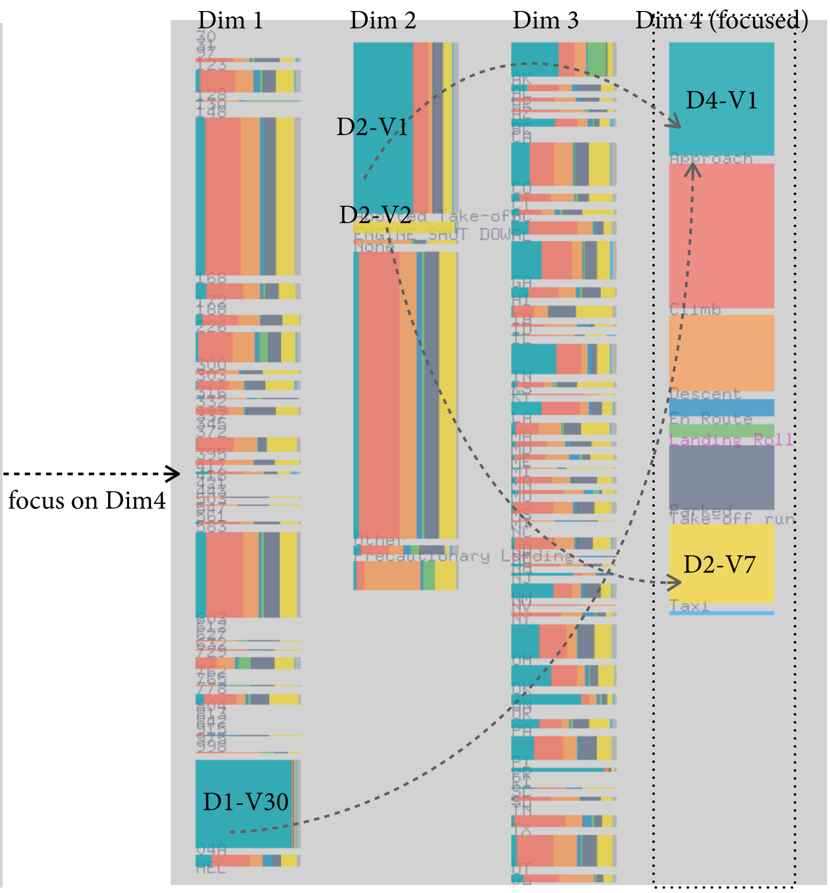


Fig. 5. Parallel Color Bar with 4 dimensions, after focusing on Dimension 4 as the color reference.

The details on interface and interaction have not been mapped out at this stage of the project. But I will stick to a minimalist implementation of my design, emphasize on the concept, without overloading it with features.

VII. Scenario of Use

Let's imagine two scenarios based on the two types of correlation detection addressed in the I. Domain section, which are *validating known correlations* and *detecting correlation outliners*.

To validate known correlations, let's suppose the Dimension 4 in Fig. 5 correspondent to different *flight phases* that an airplane have failed. (Please ignore the faint labels in the figure, they don't quite fit with our imagined scenario here.) The first block, cyan, corresponds to the *landing phase*; it takes about 1/5 of the column height, therefore we know that 1/5 of the aircraft accident happened during landing. And let's suppose that aircraft designers already know that certain *conditions of the aircraft's tire* (Dimension 1) and certain *weather conditions* (Dimension 2) will more strongly affect the chance that an accident could happen during the *landing phase*; and they want quickly validate this knowledge.

To do so, they set Dimension 4 is the focused dimension, so the rest of dimensions (Dimensions 1, 2, 3) will be "painted" accordingly. Because our designers also know that, specifically, *Bridgestone* tires (Block D1-V30) and *raining* weather conditions (Block D2-V1) are *known* to correlate/cause accidents during the *landing phase*. So they quickly locate these two blocks; and to confirm their hypothesis, both blocks are dominated by cyan color, which correspondent to *landing phase*, therefore their knowledge are validated. Further more, they could even mouse-over the blocks to obtain more detailed information (percentage etc.) In this scenario, the ability to *quickly* view and navigate across multiple dimensions is the key.

The second scenario including discovering unexpected correlations. Let's suppose the focused dimension is still on Dimension 4. Now, the user quickly sweep over the other dimensions, *with no particular interest in mind but only cognitively focusing on color.* Here, one particular block stands out because it's all yellow (D2-V2), and it may possibly contain information that is priorly unknown. This supports the kind of passive, bottom-up way of analysis.

VIII. Implementation

Primary language would be C++. I will not use any visualization toolkits (bespoke is the funnest). The design is simply enough just to draw with OpenGL. For the rather complicated event handling/interactivity I would use Poco and openFrameworks, in which Poco provide a very minimalist *Observer* event listening pattern, and openFrameworks provides a cross-platform windowing environment. For the most important aspect of data handling, I would use SQLite for its simplicity and speed.

IX. Milestones

Nov 15. Finishing categorical data implementation
Nov 30. Finishing numerical data implementation
Dec 7. Finishing interface/interaction
Dec. 14. Presentation

Reference

[1] Inselberg et al; Parallel Coordinates: A Tool for Visualizing Multi-Dimensional Geometry, 1990.

[2] Holten, D; Hierarchical edge bundles: Visualization of adjacency relations, 2006.

[3] Friendly, M; Mosaic displays for multi-way contingency tables, 1994.

[4] Bendix, F; Parallel sets: Visual analysis of categorical data, 2005.

[5] Cleveland et al; Parallel sets: Visual analysis of categorical data. 1984

[6] Christ, RE; Review and analysis of color coding research for visual displays, 1975.