Visualizing Graduate Admissions Data
CMPS 161: Final Project

Aaron Doubek-Kraft
adoubekk@ucsc.edu

March 20, 2017

Abstract

The unusually large applicant pool to the Computer Science graduate program at UCSC in 2017 introduced a problem of scale to the admissions process. By applying multivariate visualization techniques to applicants' admission data, I aim to identify correlations in the data that could be applicable to the selection process in the future, such as whether test scores correlated to interest/strength a given research area, or similarities between accepted applicants. I employ a common information visualization technique known as the parallel coordinate plot. A brief overview of this technique is provided, as well as descriptions of several common methods to improve its readability and usefulness that are implemented in this project.

1 Introduction

In 2017, there were nearly 1100 applicants to the UC Santa Cruz Computer Science graduate program. Given the unusually large size of this applicant pool, making sense of any large-scale trends simply by looking at their records in a table becomes an intractable problem. In addition to the high volume of applicants, the students' records contain a large number of potentially significant variables to be analyzed. This includes quantitative data, such as test scores and GPA, and qualitative data, such as the applicants' research interests, countries of origin, and whether or not the student was admitted. In this paper, I develop visualizations to identify correlations in this data that could be relevant to the selection process. In particular, I analyze the correlation between research interest and relative performance on the graduate entrance exams,

2 Methods

The large number of variables and the high volume of records to be analyzed makes this a classic multivariate visualization problem, and so I take a relatively standard approach: the parallel coordinate plot. In a typical graph, the number of variables available to be displayed is limited to two or three, the number of spatial dimensions, so only correlations between a relatively small number of variables may be analyzed at a time (for example, in a scatter plot). The parallel coordinate plot generalizes the graphical approach to an arbitrary number of axes, allowing correlations to be analyzed across a large number of variables.

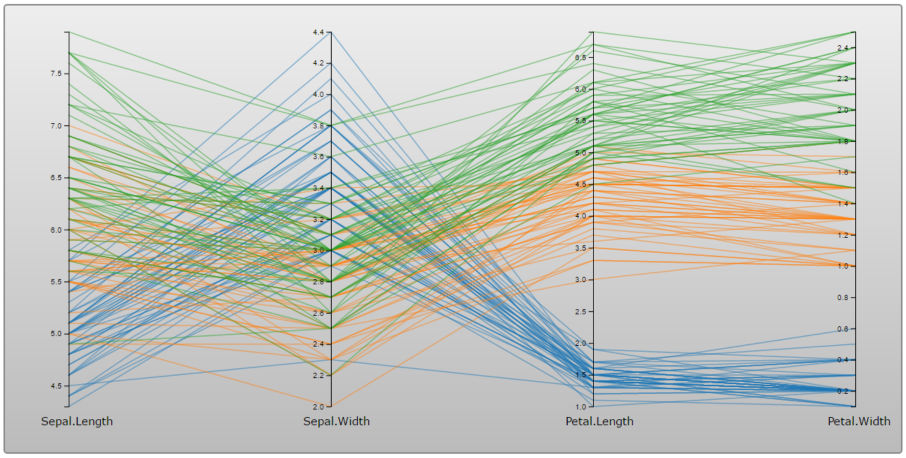

Figure 1: Visualizing Edgar Anderson's Iris dataset. In this example, color is mapped to species of iris.

As an example, I present a standard test case for multivariate visualization: the iris dataset.[9] Even in the absence of any specific knowledge about irises, some trends are immediately apparent. First, petal length and petal width are clustered for a given species, and they appear to be directly correlated: irises either have large petals or small petals. Similarly, sepal length and sepal width are also clustered, though not as strongly as petal size, and they appear to be inversely correlated (in case the reader is curious, the sepals are the green parts outside the flower that protect the petals, a fact I had to look up). Using the high dimensionality, we can compare axes across the chart, and we can see that sepal length is also correlated to petal width. Now, the strength of this approach becomes clear: it would have taken four or five lower-dimensional charts such as scatterplots to reach the same conclusions that are presented here in one condensed figure. I leave analyzing the significance of these trends to the botanists.

The parallel coordinate plot, in its most basic form, also has the advantage of simplicity: straight lines connecting parallel axes are not difficult to implement, especially given the powerful scaling and rendering functionality of the d3.js library[3] used in this project. The challenge is in scaling the visualization up to larger datasets. The iris dataset consists of only 150 records, while the graduate admissions data has almost 1100, so approximately seven times the number of lines will be drawn on the plot. Without some additional adjustments, the plot will be too visually cluttered to be of any use. One simple technique to improve readability is to map the color of the lines to their group membership, as color is mapped to species in the iris example. I will discuss additional techniques in the Implementation section.

3 Implementation

3.1 Technical

This project uses d3.js 4.7.3 to handle the rendering of the visualizations (modules: d3-axis, d3-scale, d3-selection, d3-brush, and d3-shape), jQuery.js for DOM manipulation and user input, w3.css as a style framework, and Python scripts for dataset processing tasks. The data is read into Record objects, each of which contains two native JavaScript maps: one from keys to values for qualitative data (defined as "sets" or "groups" in the code) and one from keys to quantitative data (defined as "data"). These record objects also define getter and setter functions, although this is mainly a matter of convenience as JavaScript does not provide a robust system of encapsulation.

3.2 Developing Visualization

In this section, I will discuss the strategies implemented in this project to improve readability of parallel coordinate plots of large datasets. While the primary objective here was analysis of the graduate admissions data, the techniques used here, and in fact the interface itself, could be used to visualize any multivariate dataset stored as a csv file. In general, however, csv is a relatively loose standard[2], so any truly generic application is nearly impossible.

3.2.1 Color Mapping

As the interface was designed to be a relatively generic approach to multivariate data, the program will parse potential relevant groups from the dataset itself and color-code them according to group membership. If the dataset contains multiple potential groupings, as the graduate admissions data does, the interface also allows the user to decide which grouping to use. This facilitates comparisons between subsets of the data, as well as identifications of correlations within a particular group.

3.2.2 Highlighting

If the user is interested only in a certain subset or subsets of the data, those may be selected, and all other subsets will be grayed out. This further emphasizes correlations (or the lack thereof) within this subset.

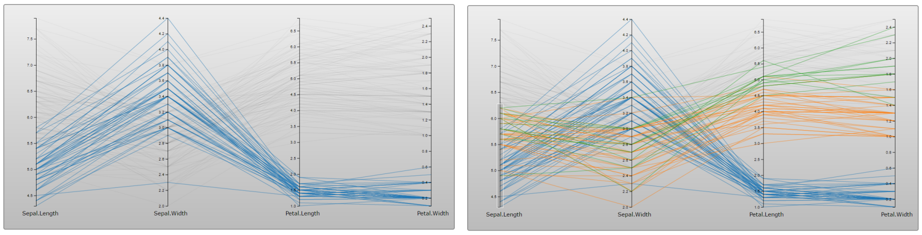

Figure 2: Left: Iris dataset, with only species *setosa* highlighted. Right: same dataset, with a brush applied for low sepal width.

3.2.3 Brushing

Brushing involves selecting only those records which fall in a particular range on a given axis [1, 7], and highlighting the associated polylines. This technique is particularly well suited to identifying correlations in larger datasets, because it allows the user to identify clusters within the data, and track those clusters across the many variables.

The algorithm used to determine which lines have been selected is described as follows:

```
Data: d3.js brush objects, which contain currently brushed extent on a
      given axis, and an array of records
Result: an array of booleans indicating whether the line corresponding
        to that record has been selected
n = number of records;
brushed = array of booleans with size n, initialized to true;
for each axis do
    extent = selection area of brush on this axis;
    for each record do
        datapoint = value of record on this axis;
        if datapoint within extent then
            brushed[this record] = false;
        end
    end
end
return brushed
```

3.2.4 Axis Selection

The visualization interface allows the user to select what axes to be displayed on the plot. While most of the strategies discusses here focus on highlighting clusters or trends in the data, this was implemented as a way to reduce the scale of the problem. For example, variables that take on only a small range of discrete values are typically not suited to this type of visualization[7], so this tool can simply eliminate such variables from the view.

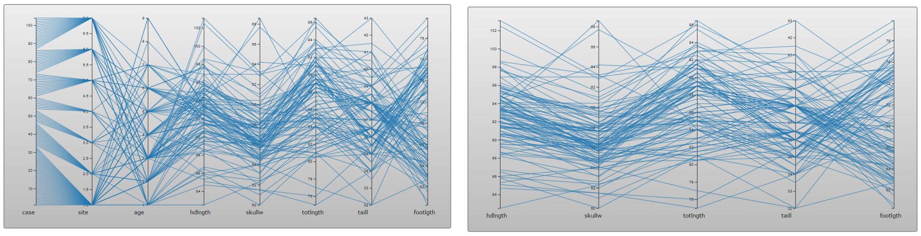

Figure 3: Left: Possum dataset[9], with discrete variables on left axes. Right: same dataset, with discrete variables eliminated using axis selection.

3.3 Dataset

The dataset consists of 1097 records, each representing an applicant to one of the UCSC computer science graduate programs. The fields from the provided dataset that were utilized to create the visualizations are described as follows:

Field	Description
ID	Applicant identification number
Tags	Filters and Professors' interest in applicant
Degree	Masters or PHD
Research Interests	Applicant's list of research interests
UG GPA	Undergraduate GPA
UG GPA SCALE	Scale for undergraduate GPA
GRAD GPA	Graduate GPA
GRAD GPA SCALE	Scale for graduate GPA
GREV	GRE - Verbal Reasoning
GREQ	GRE - Quantitative Reasoning
GREA	GRE - Analytical Writing
TOEFL	Test Of English as a Foreign Language
IELTS	International English Language Testing System
For/Dom	Domestic, or Country of Origin
Admitted	Yes or No

Since the applicants come from all around the world (in fact, the vast majority of applicants were foreign), one obstacle to drawing objective quantitative comparisons between groups of students was the fact that not all students have taken the same tests, and GPA is scaled differently in different countries. Normalizing the GPAs was relatively straightforward, since the dataset includes both the student's GPA and the GPA scale used, so the scores were simply normalized to 1, or in other words, the fraction of the appropriate total possible GPA. Additionally, a small number of the reported GRE scores were far higher the actual maximum score on the exam[5], so these were simply thrown out. Finally, there are two different English as a Foreign Language exams, the TOEFL[11] and the IELTS[6]. These two were merged into a single EFL Index, again by normalizing to the maximum score on each exam. However, these two exams are score quite differently: TOEFL reports a score in the range 0-120, while IELTS assigns a "band" from 0-9 based on proficiency. A more representative comparison might be to fit each set of scores to a distribution, but there were relatively few IELTS scores, so this approach may also have introduced some bias as well.

Additionally, once I began working with the dataset, I began to suspect that the trend of domestic students outperforming foreign students on the Verbal Reasoning and Analytical portions on the GRE may have been overshadowing any more subtle trends relating to their research interests. Therefore, I also included a normalized GRE score for each category, defined as the simple distance from each score to the mean score of applicants from that country (with negative meaning below average). To summarize, the fields I added to the dataset are defined as follows:

Field	Description
UG GPA (Normalized)	$\frac{\text{UG GPA}}{\text{UG GPA SCALE}}$
GRAD GPA (Normalized)	$\frac{\text{GRAD GPA}}{\text{GRAD GPA SCALE}}$
EFL (Normalized)	$\frac{\text{Score}}{\text{Max Score}}$, for TOEFL and IELTS
GREV (N)	Score − Mean Score for Country of Origin
GREQ (N)	Score − Mean Score for Country of Origin
GREA (N)	Score − Mean Score for Country of Origin

4 Results

Once I began working with the graduate admissions dataset, it become clear that the set was simply too dense to be interpreted as a whole. In my original implementation, the lines that were not selected were simply grayed out. While this made for a nice visual effect on smaller datasets, on the graduate admissions dataset, the plot was simply too densely populated, and even the grayed out lines became a visual distraction and hindrance. For the graduate admissions data, this effect transparency of these lines was further increased, almost to the point of removal. In addition, all applicants that had been tagged with ”Fail First Filter” were thrown out, which reduced the problem size by roughly half. All visualizations in this section were generated with this reduced dataset.

4.1 Admitted Students

As might be expected, the students who gained entrance overwhelmingly scored above average on each of the GRE sections (average being 0 on each of the GRE* (N) scales). While many of those who were not admitted also scored above average, the admitted group mostly excludes those who scored below average. This visualization also entirely omits those who met the ”Fail First Filter” criteria, which often excluded students with very low GRE scores.

4.2 Research Interests

I attempted to identify correlations between particular research interests and improved performance on specific areas of the GRE. For example, I hypothesized that interest in Natural Language Processing and Text Analytics might correspond to above average scores on the Verbal Reasoning and Analytical Writing portions of the exam, whereas interest in Data Analytics or Machine Learning might correspond to above average performance on the Quantitative Reasoning section. However, I discovered that there were no such trends to be

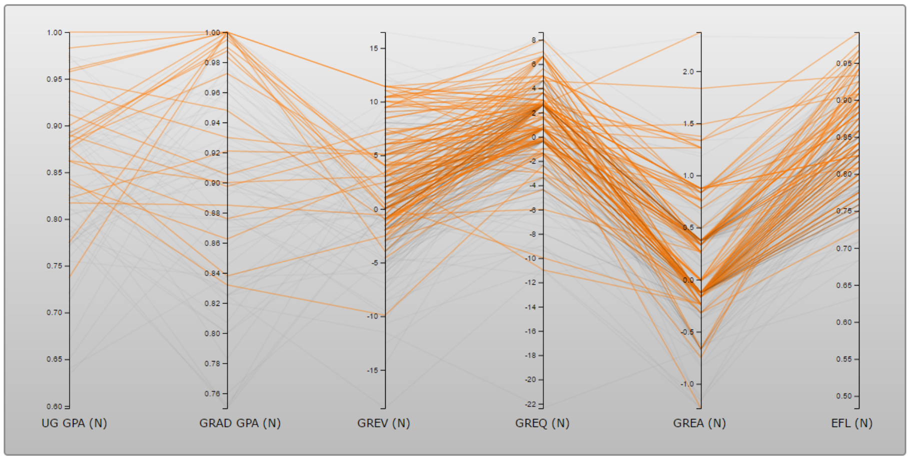

Figure 4: Students offered admission to the UCSC graduate computer science program.

found. As the example plot in figure 5 shows, students interest in Software Engineering, Data Analytics, Natural Language Processing, and Computer Games and Digital Media showed no significant difference in performance on any of the 3 exams. The results were similar for all grouping by research interest that I tried.

4.3 MS vs. PHD

I expected to find no major difference between Master's and PHD applicants, but I was surprised to find that MS applicants, on average, performed better on the GREQ and GREA than their PHD counterparts, as figure 6 shows.

4.4 Previous Grad School Attendees

This plot demonstrates an alternative use of the brushing feature. Brushing only the graduate GPA axis effectively selects those students who have previously attended graduate school. These students are also somewhat below average for GREQ and GREA, which while surprising taken on its own, actually follows logically from the results of the MS vs. PHD section, since the majority of those students with previous graduate education are most likely applying to the PHD program.

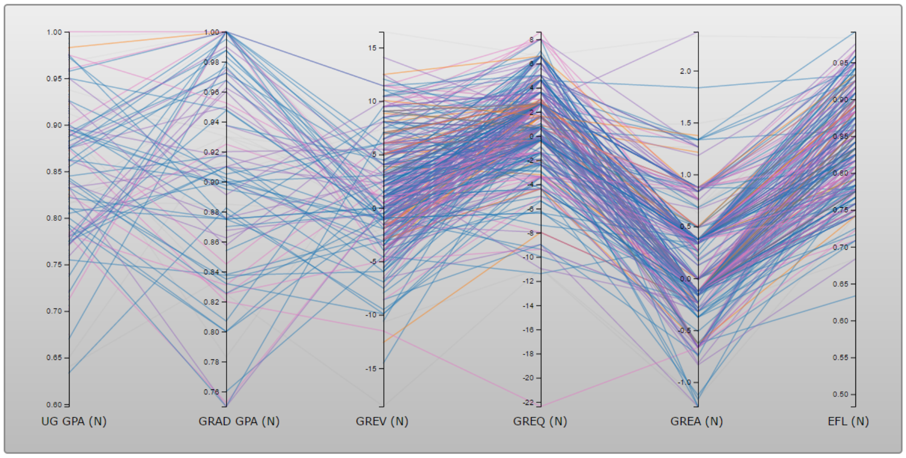

Figure 5: Grouping based on research interest. Blue: Data Analytics; Pink: Software Engineering; Purple: Natural Language Processing; Orange: Computer Games

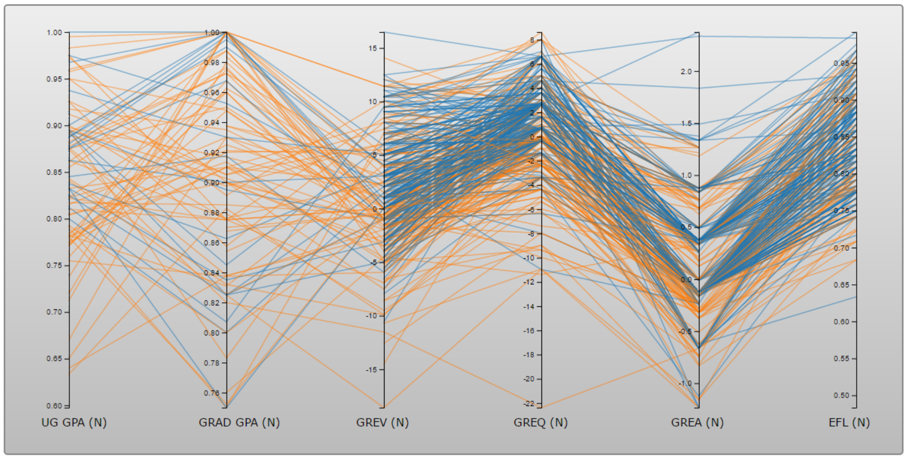

Figure 6: Plot with grouping based on application to the MS or PHD programs. Blue: MS; Orange: PHD

5 Related Work

5.1 Parallel Sets

A variation of parallel coordinates emphasizing aggregate group behavior. Implemented as a desktop Java application by Kosara and Ziemkiewicz.

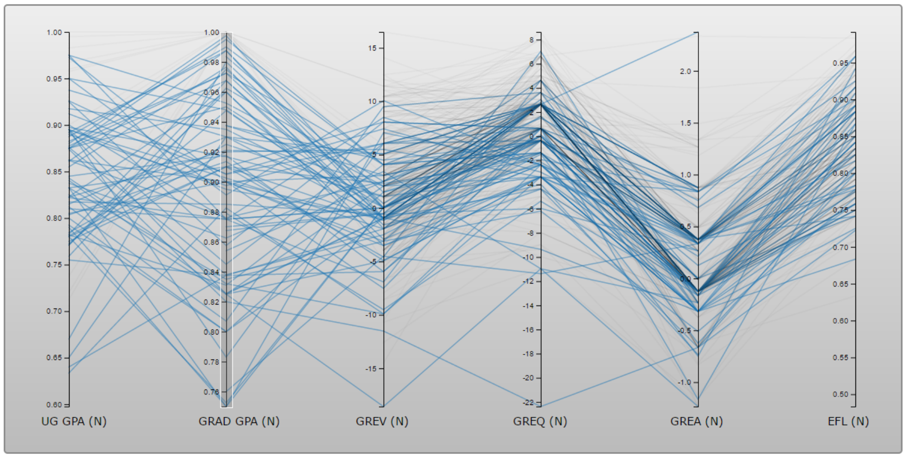

Figure 7: Plot brushed to include only those who had previously attended a graduate school

5.2 RadViz

A radial coordiante visualization, essentially amounting to star plots superimposed on top of one another. Well suited to identifying cluster behavior. [4]

6 Conclusion

There is a great deal of potential for improvement of this interface. In order to be a useful generic tool, many more of the visual attributes of the graph could be customizable, including the colors corresponding to particular groups, whether unselected lines are grayed out or hidden, how and where the labels are placed, etc. In addition, there are several features I planned to implement but couldn't fit in before the deadline. First, I planned to allow switching from the quantitative ”data” category to the qualitative ”sets” category, as some discrete variables (such as store number, number of letters received, etc.) could be more useful as groupings than as plotted data. Another common feature of this type of vis involves allowing the axes to be reordered, which can in some cases make correlations more apparent. Finally, one useful feature for dealing with large datasets would have been to be able to apply multiple filters based on grouping, so the problem size could be further reduced to correlations of groups within groups.

References

[1] Bostock, Mike. Parallel Coordinates Example [Internet]. d3.js; Available from http://mbostock.github.io/d3/talk/20111116/iris-parallel.html

[2] Comma Separated Values (CSV) Standard File Format [Internet]. Edoceo. Available from http://edoceo.com/utilitas/csv-file-format

[3] d3.js: Data Driven Documents. Available from https://d3js.org/

[4] Fayyad, Usama; Grinstein, Georges; Wierese, Andreas. Information Visualization in Data Mining and Knowledge Discovery. London: Academic Press; 2002 [cited 2017 March 19]. Available from https://books.google.com/books?id=rYFvnyPRwkgC&lpg=PR5&dq=multivariate%20visualization%20information&lr&pg=PR4#v=onepage&q&f=false

[5] GRE Scores [Internet]. Available from https://www.ets.org/gre/revised_general/scores/?WT.ac=grehome_grescores_150213

[6] How IELTS is scored[Internet] https://www.ielts.org/about-the-test/how-ielts-is-scored

[7] Kosara, Robert. Parallel Coordinates [Internet]. Eagereyes; 2010. Available from https://eagereyes.org/techniques/parallel-coordinates

[8] Kosara, Robert; Ziemkiewicz, Caroline. Parallel Sets [Internet]. Eagereyes; Available from https://eagereyes.org/parallel-sets

[9] R Datasets [Internet]. Available from: https://vincentarelbundock.github.io/Rdatasets/datasets.html

[10] Telea, Alexandru C. Data Visualization Principles and Practice. 2nd Edition. Boca Raton(FL): CRC Press; 2015.

[11] TOEFL iBT Test Scores [Internet]. https://www.ets.org/toefl/ibt/scores