# PARALLEL COORDINATES VISUALIZATION OF LARGE DATA

Suchismitha Vedala
svedala@uh.edu
UH ID: 1470929
University of Houston
Department of Computer Science

## ABSTRACT:

The ability to visualize data often leads to new insights. Data that is more than three dimensional must be visualized as a series of projections or transformed into some other representation which usually causes a loss of details or many outliers. Representing data as such can also often lead to fuzziness and confusion. Parallel coordinates allow us to visualize large data with multiple axes in two dimensions without a loss of information. In this project, we discuss the use of parallel coordinates to visualize a large data set consisting of the statistics of applications which are widely used. Further, we discuss the implementation of the project and methodologies used for the dynamic implementation of the data using parallel coordinates

## 1.INTRODUCTION

Visualization is the fundamental process for exploring and dissemination of information through visual aids such as diagrams, sketches, photographs, charts, video, animations etc., thus improving the quality of comprehension, memory and inference. The strength of visualizations lies in the design principles which connect the visualizations with viewer's perception and cognition of underlying information that is conveyed. The key to good visualizations use the best practices to either emphasize the relevant information or de-emphasize irrelevant information.

Visualizing information that has more than 3 dimensions is not understandable to the human eye in the 2 Dimension. With the data set "Details of Top 500 Applications", we have the statistics of the names, category, rank, installations, price etc. Visualizing all this data becomes increasingly complex in a 2 Dimension. Considering various visualization techniques, we adopt parallel coordinates for our project.

Parallel coordinates are introduced by Philbert Maurice D'Ocagne in 1885. Parallel coordinates is a technique for enabling the visualization of high-dimensional data which could not be effectively visualized through the use of standard two- or three-dimensional graphs or tables. It accomplishes this by displaying each data field as a single vertical axis and laying out these axes horizontally, drawing each data point as a line that intersects each vertical axis in turn. Because humans are highly visual learners, this technique improves people's ability to analyze high dimensional data effectively.

*Figure 1: A simple parallel coordinates*

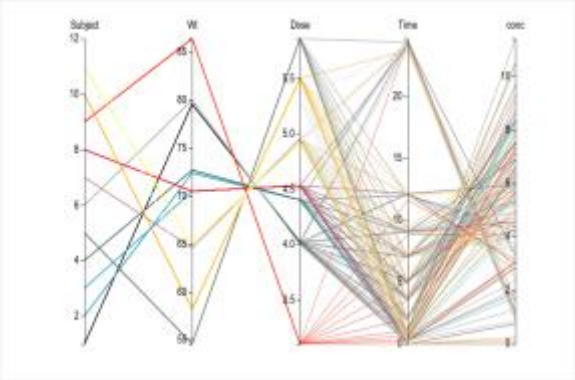


*Figure 1 shows a sample visualization example demonstrating the parallel coordinates.*

## 2.RELATED WORK

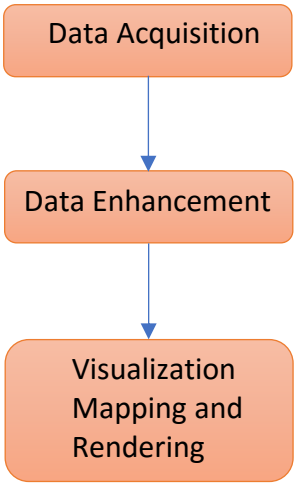
Information Visualization is popular. There are many tools, libraries that will offer help to visualize the data with minimum effort. ***Kai Chang*** has worked in this field and also gave a talk on "*Visually Exploring Multidimensional Data*" at OpenVisConf. "*Visualizing the iOS App Store"* was developed by *Sensor Tower* adapted from Kai Chang. *Jason Davies* has developed an interactive parallel visualization for the "*Titanic Survivors".*

## 3.METHODOLGY AND IMPLEMENTATION

### 3.1VISUALISATION PIPELINE:

To understand visualization, we first need to visualize how the process works. In our project, we proceed with the steps as shown in the pipeline.

*Figure2: Visualization Pipeline*




#### 3.1.1 DATA ACQUISITION

The data acquisition consists of obtaining the data set. We shall consider the data set of our project. We intend to visualize the data of top 500 most grossed applications of the year 2015. This data set is huge with many attributes such as title, verison, price_usd, rank_change_day, rank_change_week,category, content_rating, developer, store_url, votes, reviews, updated, min_instals, max_installs, This is the original data set which is obtained ad downloaded from the net.

#### 3.1.2.DATA ENHANCEMENT

From the above acquired data, we decide on which attributes to plot on the graph. The name of the applications is displayed in a column and the categories of the applications in another with the reviews, min_installs, max_installs, ratings, reviews and rank plotted as parallel coordinates.

### 3.1.3. VISUALISATION MAPPING AND RENDERING

To build the visualization of our project, we will use standard web-technologies such as HTML, CSS and JavaScript. Document Object Model (DOM) is generally used for web-applications. The basic model of these technologies is DOM based but their implementation varies for different browsers.

***HTML and CSS***

HTML stands for Hypertext Markup Language1 and it is developed by a scientist Tim Berners-Lee in 1990. The purpose was to make it to design semantics for scientific documents so that researchers at different universities can access each other's work easily. Later it laid the foundation for the World Wide Web (WWW). It is a text markup language which is used to create documents on he web. It is a hidden code in a web-browser which helps browser to create the structure and layout for a web-document. Every page on Internet contains HTML code. It is just like a wrapper used to structure contents for web-browsers by using a variety of tags ant attributes. HTML concentrates only on structure of text rather than visual appearance. To make the page visually more appealing, CSS style is applied on web page. Cascading Style sheets or CSS was invented to describe the presentation of a document written in markup language. CSS describes look and format of a document element on web-page like color, size, shape of elements, etc. CSS enables us to separate the presentation style such as layout, colors and fonts from the contents of documents.

***JAVASCRIPT***

JavaScript is a lightweight scripting language2 used for Web pages. It provides a facility to control the browsers, alter the web-document contents, etc. JavaScript manipulates the internal Document Object Model (DOM). It provides various features for web-pages such as check forms, customize graphics selection, widgets, graphics and many more. JavaScript is most widely used scripting language but it has few drawbacks also. For example, there is no namespace, import statement and access modifiers in JavaScript. To get rid of these shortcomings of JavaScript, number of other scripting languages have been developed over a time.

***VISUALISATION TOOLKITS***

To develop our project in real time, and for dynamic interaction we use visualization toolkits such as d3.js and underscore.js.

***D3.js***

D3.js (Data-Driven Documents) is a free and open-source JavaScript library developed by the Stanford Visualization Group (Mike Bostock) in 2011. D3.js uses digital data to drive the creation and control of dynamic and interactive graphical forms which run in web browsers. It is a tool for data visualization in W3C-compliant computing, making use of the widely implemented Scalable Vector Graphics (SVG), JavaScript, HTML5, and Cascading Style Sheets (CSS3) standards. It is the successor to the earlier Protovis framework. D3 emphasis on web standards gives you the full capabilities of modern browsers without tying yourself to a proprietary framework, combining powerful visualization components and a data-driven approach to DOM manipulation. D3 has the following advantages.

• It has full capabilities of modern browsers • Input data which can bind to a DOM can be of following forms: Comma Separated Value, JavaScript Object Notation, eXtensible Markup Language files

• It is fast even in case of large datasets

• Dynamic interactive visualizations can be realized.

***Underscore.js***

Underscore creates and exposes all its functionality via a single object, in global scope. This object is the titular underscore character, _. It is comparable to features provided by Prototype.js and the Ruby language, but opts for a functional programming design instead of extending object prototypes. For example, Underscore.js's _. _. each function delegates to the host environment's native for Each implementation when present, or a compatible version when absent. Underscore.js was created by Jeremy Ashkenas, who is also known for Backbone.js and CoffeeScript. . Underscore provides a little more than 60 functions that span a few functionalities. At their core, they can be classified of functions that operate on:

- Collections
- Arrays
- Objects
- Functions
- Utilities.

## 3.2. IMPLEMENTATION:

- From the acquired data set, we extract the attributes with their columns and export it to a new csv file which serves as the enhanced data set for our project.
- We begin by scripting our HTML to visualize the data on the web browser.
- The designing of the page is written in a CSS file, which is included in the HTML script.
- We include the visualization toolkits such as d3.js and underscore.js in the script source before our JavaScript file which calls the data set and performs the dynamic interaction.
- We begin the scripting of the JavaScript file by first assigning the features (color, height, width) to our interface window.
- We then extract the data from the file and calculate the attributes and assigning a scale value to each.
- We also implement ***"opacity"*** to the data which shows the percentage of opacity dynamically at any point.
- We implement the feature of exporting of data "*Export"* which at any point when clicked exports the csv data in a new tab on the browser.
- The following filters are applied to the data set.
  - Brush
  - Reorder of the axes
  - Invert the axes
  - Progressive Rendering

## 4. RESULTS AND DISCUSSIONS

The results of the implementation of our project can be visualized with the Figures 2 and 3. The format of the data of the applications is {Min Installs, Rank, Rating, Review, Max_installs}. The following operations are provided by this visualization.

- ***Keep*** : This button allows the user to select which data to be zoomed in on the interface window.
- ***Exclude***: This button allows the user to select which data to exclude on the visualization window. If chosen to remove all, it displays an error
- ***Export***: This exports the data of the plotted lines. The user can also dynamically select few applications and export only that particular data.
- ***Hide Ticks/Show Ticks***: This button allows the user to hide/show the ticks (index values) respectively. However, only one button can be selected at any point of time.

- ***Dark/Light***: This button allows the user to select the background of the visualization window. Dark corresponds to black while Light corresponds to white.
- The categories of applications consists of the list of all the categories these applications belong to.
- ***Unselect All Categories***: Clicking this button at any point while unselect all the data and thus showing no output.
- Clicking on any category on the list of categories of applications, will remove the plotted lines for all the applications corresponding to the category from the visualization plot.
- Moving the cursor on any application gives the corresponding plot line and category on the data plot.
- The ***Controls*** show the list of operations that can be performed with the axes and the plot.
- The ***Reset Filters*** button resets all the filters which were applied till then and gives the original.

*Figure 2: The User Interface of our Project (Part 1)*

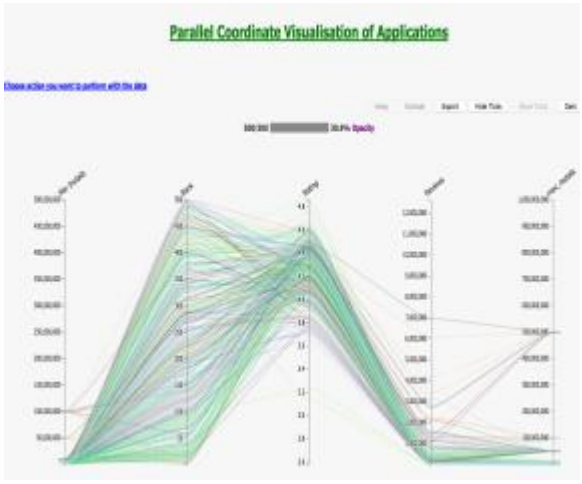
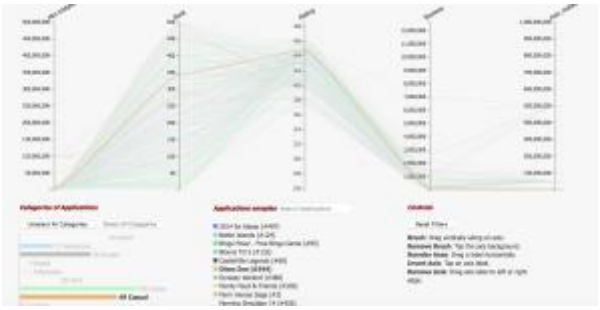



*Figure 3: The User Interface of our Project (Part 2)*



There are quite a few interesting aspects obtained from the results. The observations give us an analysis of various features and would answer many questions posed by an observer to the project

1. Moving the cursor on any one on the application samples makes it bold and the corresponding category is visualized in bold and the plot line belonging to only that application appears on the visualization window. Also index number beside the application samples is the number at which the application is present in the csv file.

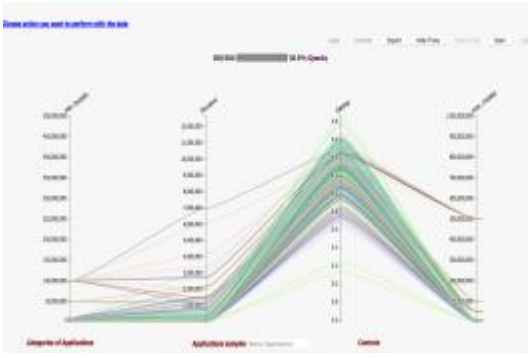
*Figure 4: Showing the index and the application samples variation by cursor movement.*

2. Searching an application in the search box gives the result of the plot line of only that application (making the window free of any other plots). This is evident with figure 5.

*Figure 5: Search box results*

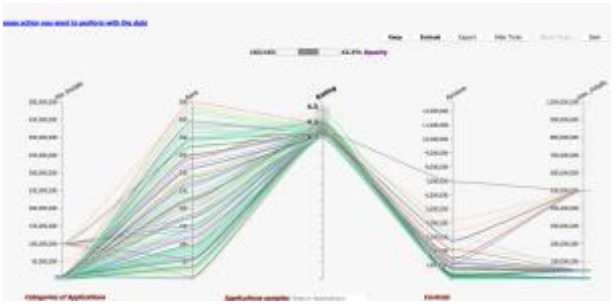
3. While dragging an axis to left or right for reordering, if we move to the extremes of the window, the axes is removed.

*Figure 6: The rank axes when being dragged to the extreme, has been removed from the interface window.*



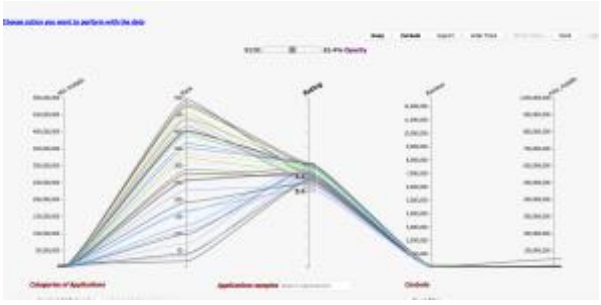
4. If the user wants to display the list of plots falling in a region of the axes, the region is selected and all the plot lines which fall in that category are displayed

*Figure 7: Displaying only specific results.*



5. With progressive rendering, we can move that region along the axes to view all the plot lines which fall in that range.

*Figure 8: Progressive rendering*



6. When we select a few and we click the keep, it zooms in the data and changes the axes indices per it.

*Figure 9: Selecting a few applications.*

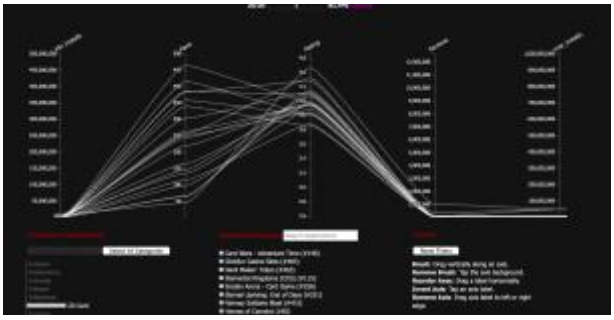
*Figure 10: The result of the selected applications on Keep button*

7. All the data can be obtained without having to look at the source file. This is possible when the user selects the export option which exports the data.

*Figure 11: Export data results*

8. For better visualization, the user can also select a darker background by clicking the dark button. This helps in visualizing plots with chosen light colors.

*Figure 12: Dark Button*



9. If the user just wants how the interface looks, without any indices on the axes, the hide ticks would be helpful. This can be reverted which the show ticks button is clicked.

*Figure 13: Hide Ticks*

## 4. CONCLUSION AND FUTURE WORK

Best efforts have been put to implement this project in the most effective and interactive manner. Based on the data set, parallel coordinates have been chosen. However, one cannot conclude which visualization technique would be best suitable to a data set at any point. The enhanced data set which we choose could be extended further to visualize with many attributes, although that would make it fuzzy

and difficult to interpret. The data set can also be tested by implementing with other visualization techniques such as scatter plots.

## 5. LIMITATIONS

There are of course limits to what can be done with parallel coordinates. When the number of data items gets very high, there is a lot of over plotting that can make it impossible to see anything. The number of dimensions on screen also needs to be below about a dozen at the same time to make sense, anything above that gets very difficult to read.

There is a lot of work in visualization on automatic axes reordering, clustering of similar axes, etc. that makes this easier for high-dimensional data. But in general, the technique simply works best for datasets with a moderate number of dimensions and no more than a few thousand records.

Also, as mentioned above, the data needs to be numerical. The technique does not work well for categorical data or data with few values per axis

## 6. REFERENCES


1. Kai Chang - Visually Exploring Multidimensional Data : https://youtu.be/ypc7Ul9LkxA
2. http://davis.wpi.edu/~xmdv/docs/vis99_HPC.pdf
3. http://www.cse.ust.hk/~huamin/eurovis08_zhou.pdf
4. https://eagereyes.org/techniques/parallel-coordinates