Big Data Visual Analytics with Parallel Coordinates

Julian Heinrich
CSIRO Digital Productivity
North Ryde, Australia
Email: julian.heinrich@csiro.au

Bertjan Broeksema
Luxembourg Institute of Science and Technology (LIST)
Belvaux, Luxembourg
Email: bertjan.broeksema@list.lu

***Abstract*—The abstract goes here.**

I. Introduction

In many contexts we find the need to explore large and complex data sets with respect to data points as well as number of variables. A technique, well known in the visualization community, for exploring high variate data sets is Parallel Coordinates [1]. Parallel coordinates visualizes many variables at the same time by drawing an axis for each variable parallel to each other. A single data point is represented by a line which crosses each axis at the value the data point selects for the variable represented by the axis. Outside the visualization community however, it seems not so widely used, despite its straight forward usage.

In this exposition we present a web-based implementation to make it accessible to and usable for a wide audience. This implementation also demonstrates the core functionality of parallel coordinates. It supports large data in terms of data points by providing progressive rendering and density based rendering features. Furthermore it shows how parallel coordinates can be extended by advanced analytics by connecting it to the R statistical computing environment through OpenCPU[**?**]. This is a work in progress, which can be accessed online at http://www.parallelcoordinates.de and serves as demonstration for potential implementation of more targeted applications.

II. Analytics

Visual analytics has been defined as the "combination of automated analytisis techniques with interactive visualizations for an effective understanding, reasoning and desicion making on the basis of very large and complex data sets" [**?**]. To this end we demonstrate how to combine a parallel coordinates visualization with R. R is both a language and an environment for statistical computing[**?**]. It is widely used in a variety of domains, among which bioinformatics is a prominent one. Combining parallel coordinates, therefore brings a wide range of analytical approaches.

A. OpenCPU

OpenCPU is a system which makes R packages available through an HTTP API. For example, in our demo application we provde a clustering function, which is by posting to: http://local/ocpu/library/pacode/R/pacode.kmeans. Arguments to a function are transmitted in the post body in json format. As a reponse on calling such a function, OpenCPU sends back a session object. This object can either be used to get the actual function output in a variety of formats, amongst which json. Or, it can be used to chain functions, by passing a session key as arguments to another function call.

OpenCPU can be run from within R, however a more scalable approach is running the dedicated OpenCPU server.

The additional light-weight OpenCPU JavaScript library makes connecting R and a web front end straight forward.

Provides caching Can be made scalable using SparkR (distributed R)

B. Clustering

R Provides many clustering algorithms out of the box A thin wrapper is required to provide a consistent API for the front-end

C. Dimensionality reduction

Like clustering, many dim. red. techniques available in R. Widely used analytical approach to search for structure E.g. PCA

III. Rendering

In order to keep the system responsive even for large datasets, our system supports WebGL where available. Using hardware-accelerated graphics greatly improves rendering speed over canvas- or SVG-based rendering and allows for the use of custom shaders to implement advanced rendering techniques for parallel coordinates such as progressive rendering [2] and continuous parallel coordinates [3].

A. Progressive rendering

B. Density

Overplotting is one of the main challenges with parallel coordinates: The view quickly becomes cluttered as the datasets grow. One possible clutter-reduction method is to compute and render the *density* of lines instead of individual lines. This approach is essentially the same as computing a kernel-density estimate for histograms or scatterplots. Figure **??** shows an example of a continuous parallel-coordinates plot with Gaussian kernels and a kernel-bandwidth of $\sigma = 0.001$.

IV. Case study

V. Conclusion

The conclusion goes here.

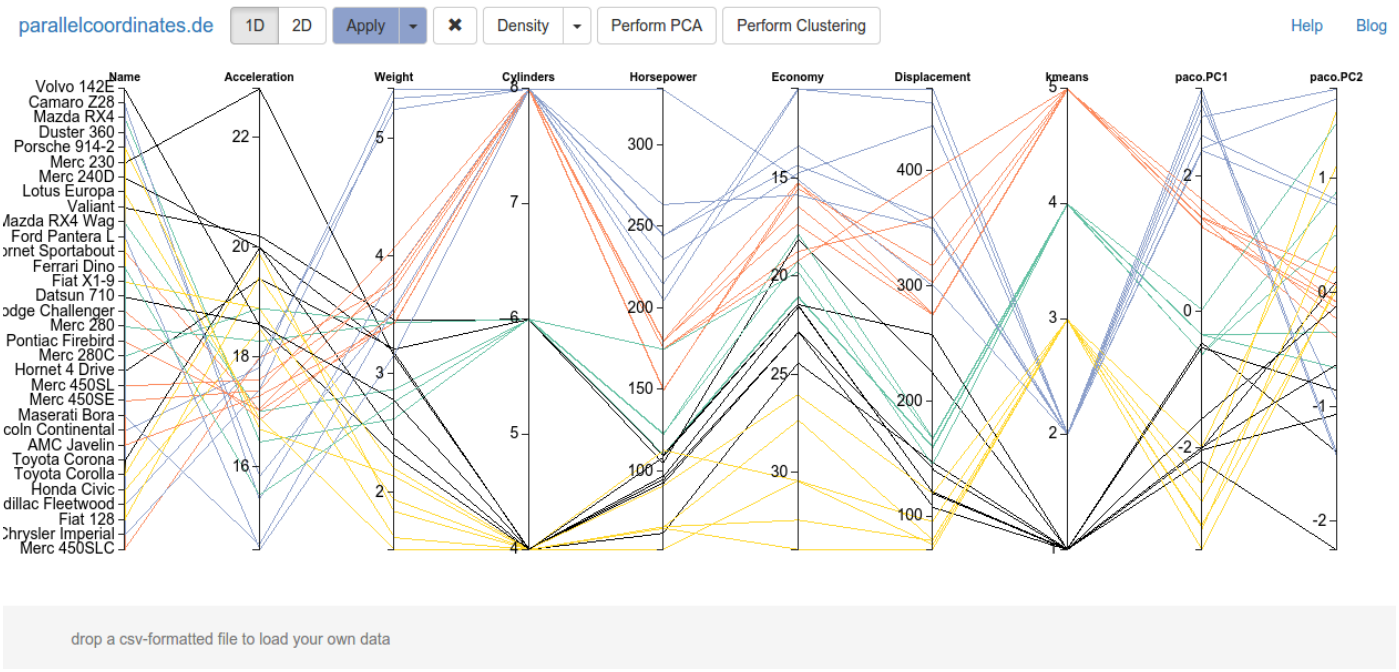

Fig. 1.

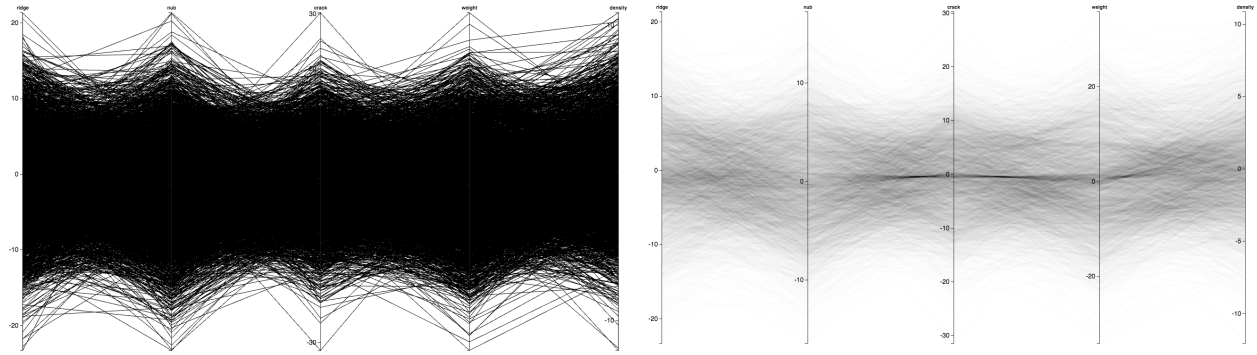

Fig. 2. In large datasets, important patterns may be obscured by the dominating feature. In this example, the envelope of lines in the left image suggests a Gaussian distribution for the data. The kernel-density estimate shown in the right image reveals the peak density hinting at a hidden cluster of lines.

ACKNOWLEDGMENT

The authors would like to thank...

REFERENCES

[1] A. Inselberg, *Parallel Coordinates: Visual Multidimensional Geometry and Its Applications*. NY, USA: Springer, 2009.

[2] J. Heinrich, S. Bachthaler, and D. Weiskopf, "Progressive Splatting of Continuous Scatterplots and Parallel Coordinates," *Computer Graphics Forum*, vol. 30, no. 3, pp. 653–662, 2011.

[3] J. Heinrich and D. Weiskopf, "Continuous Parallel Coordinates," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 6, pp. 1531–1538, 2009.