Big Data Visual Analytics with Parallel Coordinates

Julian Heinrich
CSIRO
Sydney, Australia
Email: julian.heinrich@csiro.au

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

***Abstract*—We present a web-based implementation of parallel coordinates suitable for big data visual analytics. While being easily accessible through web-browsers, the system supports advanced analytics on the server using R as well as density-based rendering on the client with support for hardware accelerated graphics. A prototype implementation is available at parallelcoordinates.de.**

I. Introduction

In many contexts the need exists 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.

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 density based rendering. Furthermore it shows how parallel coordinates can be complemented with advanced analytics by connecting it to the R statistical computing environment[2] through OpenCPU[3]. This is a work in progress which serves as demonstration for potential implementation of more targeted applications. It can be accessed on-line at:

- http://www.parallelcoordinates.de
- https://github.com/julianheinrich/parallelcoordinates.de
- https://github.com/bbroeksema/pacode

II. Analytics

Visual analytics has been defined as the "combination of automated analysis techniques with interactive visualizations for an effective understanding, reasoning and decision making on the basis of very large and complex data sets" [4]. 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 with R, therefore gives a powerful combination to interactively analyze complex data sets.

A. OpenCPU

OpenCPU is a system which makes R packages available through an HTTP API. Arguments to a function are transmitted in the post body in json format. As a response 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. Most of the low-level complexity of sending requests and retrieving results is taken away by the additional light-weight OpenCPU JavaScript library.

OpenCPU can be run from within R, however a more scalable approach is running the dedicated OpenCPU server. OpenCPU server, which can be installed on Linux servers provides caching. The caching mechanism assumes the functions in an R package to be pure functions, i.e. they have no side-effects. This allows for caching arguments and results, avoiding recalculation of results when the arguments to a function are the same as an earlier call. Further scalability when using OpenCPU can be achieved by writing the R code using the SparkR package. The SparkR package provides a front-end to Apache Spark, a large-scale data processing engine allowing for distributed computing on clusters.

B. Clustering & Dimensionality Reduction

A common way to find structure in complex high-variate data sets is by performing dimensionality reduction. Many dimensionality reduction techniques are available in the R package ecosystem. We demonstrate the use of dimensionality reduction techniques by providing a wrapper around principal component analysis (PCA). The application allows for selecting variables on which PCA is to be performed, and the number of principal components to be added to the parallel coordinates. Adding principal components to the parallel coordinates allows the user to link the more abstract principal components to concrete variables. In Fig. 1, we can see for example that the first principal component is correlated to weight, displacement and economy (negatively).

Another common task in exploratory analysis is clustering in order to find groups of similar data items. Like with dimensionality reduction, the R package ecosystem provides a number of clustering techniques. We demonstrate the use of clustering techniques by providing a wrapper around K-means clustering. The application allows for selecting variables on which the clustering is to be performed and the number of

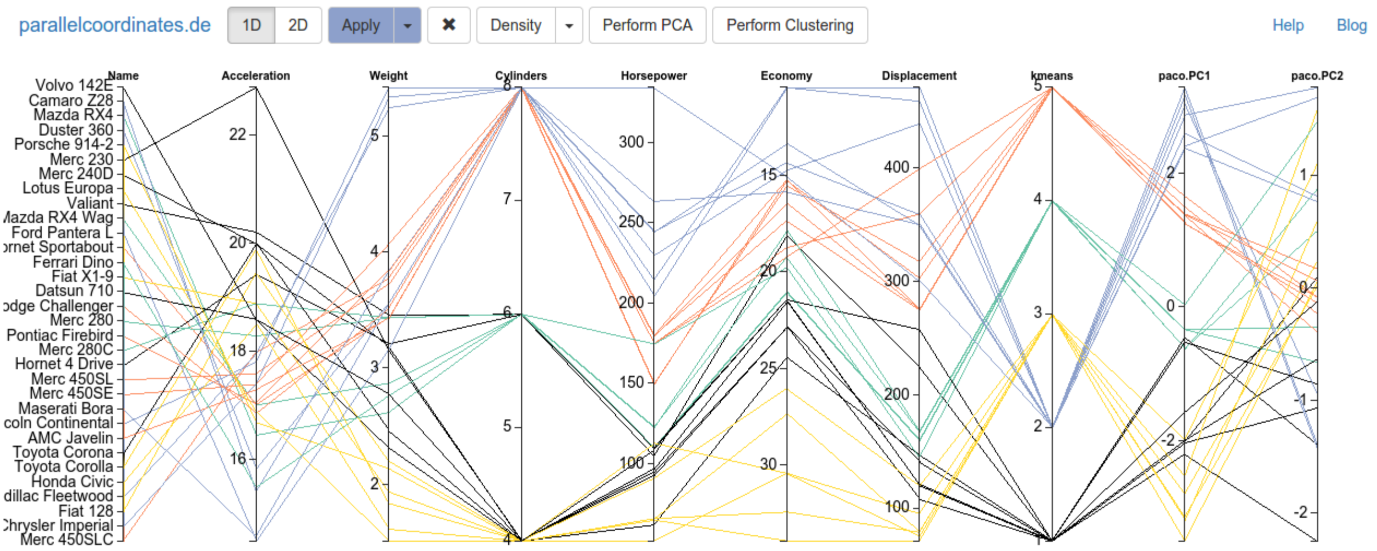

Fig. 1. Parallel coordinates, showing two principal components and a k-means clustering results, both calculated in R. Following the colored patterns we see that in this particular instance the clustering captures the first principal component quite well.

clusters that have to be returned. In Fig. 1, we can see for example that the a clustering with 5 clusters captures quite well the structure found in the first principal component.

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 [5] and continuous parallel coordinates [6].

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. By using *line-kernels*, this approach is the same as computing a kernel-density estimate for histograms or scatterplots. Figure 2 shows an example of a continuous parallel-coordinates plot with Gaussian kernels.

IV. Conclusion

We have presented a web-based system which shows how to combine parallel coordinates with advanced analytical features. The system combines web based visualization techniques, using WebGL for optimal performance, with the R analytical framework through OpenCPU. The system and its source code is made publicly available to stimulate development of application domain specific applications, using this approach.

References

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

[2] R Development Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2008, ISBN 3-900051-07-0. [Online]. Available: http://www.R-project.org

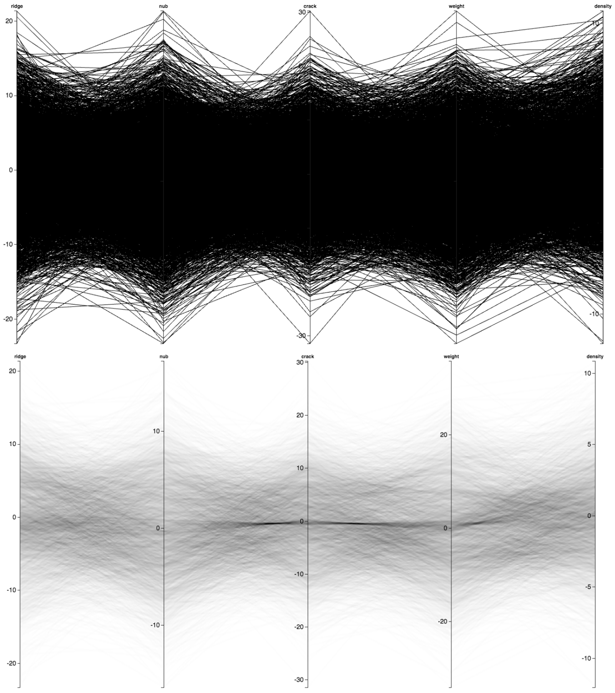

Fig. 2. In large datasets, important patterns may be obscured by the dominating feature. In this example, the envelope of lines in the top image suggests a Gaussian distribution for the data. The kernel-density estimate shown in the bottom image reveals the peak density hinting at a hidden cluster of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

[3] J. Ooms, “The opencpu system: Towards a universal interface for scientific computing through separation of concerns,” *arXiv:1406.4806 [stat.CO]*, 2014. [Online]. Available: http://arxiv.org/abs/1406.4806

[4] D. A. Keim, J. Kohlhammer, and F. Mansmann, Eds., *Mastering the information age : solving problems with visual analytics*, 2010.

[5] 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.

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