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 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, analysts need to explore large and complex data sets, both with respect to data points as well as number of variables. Parallel coordinates [1] is a technique for the visualization and explorative analysis of high-dimensional data, which visualizes many variables at the same time by drawing an axis for each variable parallel to each other. A single data point is then represented by a line crossing each axis at the value of the respective variable. While parallel coordinates are well-known in the visualization community, many other individuals from the broader data science community are not familiar with this concept. We believe that one reason for this lies in the fact that parallel coordinates are not easily accessible to the general public.

To fill this gap, we present a publicly available, web-based implementation of parallel-coordinates with support for server-side analytics and advanced rendering techniques. Hence, in addition to demonstrating the core functionality of parallel coordinates, our implementation is suitable for large data sets by providing density-based rendering [2], complemented with a connection to the R statistical computing environment[3] via OpenCPU[4]. While this work is still in progress, it serves as a demonstration of a potential implementation of more targeted applications. We are planning to add more features such as progressive rendering [5] or angular brushing in the near future. The implementation is freely available and can be accessed on-line at www.parallelcoordinates.de.

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" [6]. To this end we demonstrate how to combine a parallel coordinates visualization with R. R is a well-known language and environment for statistical computing that allows easy extension via a package system that allows us to harness its support for advanced analytics and combine it with interactive parallel coordinates to analyze complex data.

A. OpenCPU

The connection to R is possible via OpenCPU, a system which makes R packages available through an HTTP API. While OpenCPU can be used from within a local R session, a more scalable approach is to run a dedicated OpenCPU server, which allows for caching arguments and results of function calls and thus avoids potentially costly recalculations. Further scalability when using OpenCPU can be achieved by writing 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 multivariate data sets is by performing dimensionality reduction. 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 the user wants to perform a PCA, resulting in a user-specified number of principal components to be added to the parallel coordinates as additional axes. This allows the user to link the more abstract principal components to concrete variables. In Figure 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 clusters that have to be returned. In Figure 1, we can see for example that 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 [2].

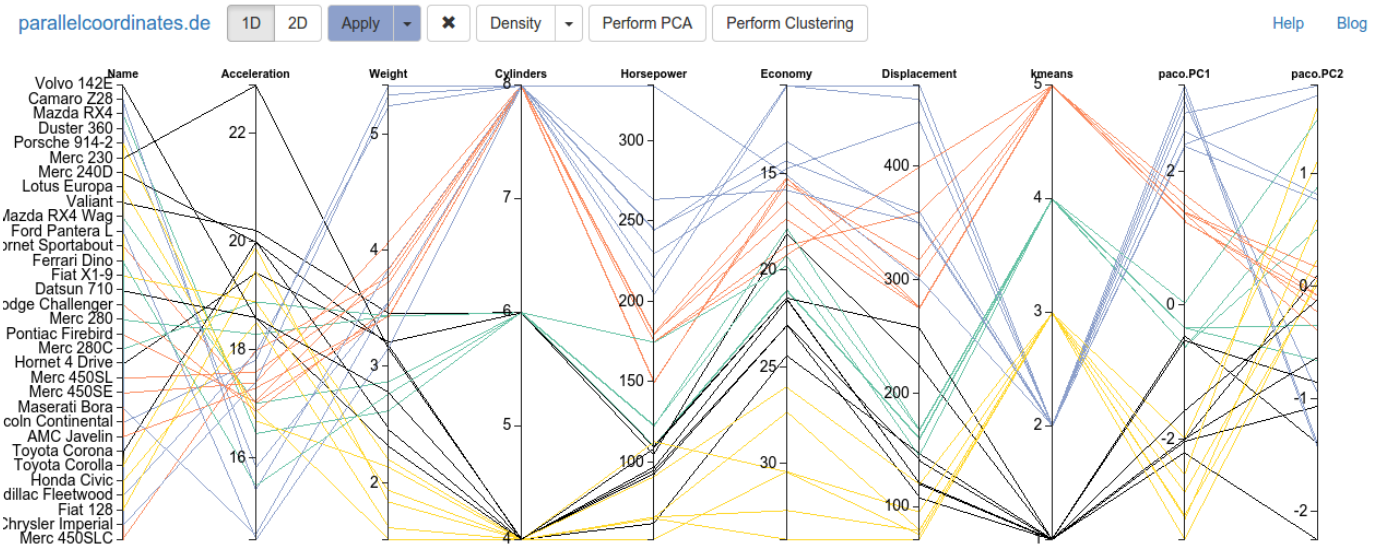

Fig. 1. Parallel coordinates, showing two principal components and a k-means clustering results, both calculated in R. Following the colored patterns we see that in this particular instance the clustering captures the first principal component quite well.

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.

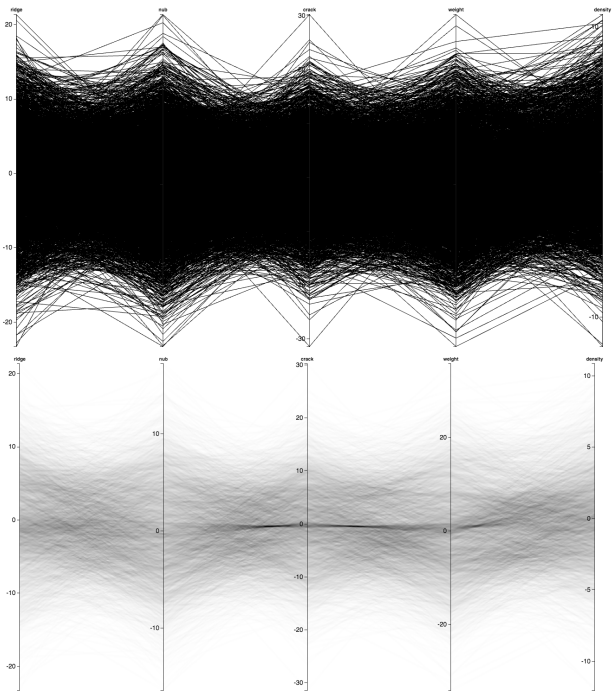

Fig. 2. In large datasets, important patterns may be obscured by the dominating feature. In this example, the envelope of lines in the top image suggests a Gaussian distribution for the data. The kernel-density estimate shown in the bottom image reveals the peak density hinting at a hidden cluster of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

IV. Conclusion

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

[3] 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

[4] 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

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