2018 IEEE 14th International Conference on e-Science

SPOT: Open source visual data analytics platform for high-dimensional scientific data

Diblen F[1], Attema J.J.[1], Bakhshi R.[1], Stienen, B.[2], Hendriks, L.[2] and Caron, S.[2]

One of the most challenging tasks in the field of information visualization is to design tools for high-dimensional scientific data. Existing visual data analytics tools either require expert knowledge (e.g. programming) or they are commercial tools such as Dash [1], Spotfire [2] and Tableau [3].

Spot provides a user-friendly, interactive data exploration environment for high-dimensional data sets [4]. The main focus of Spot is on scientific use, with the aim to facilitate open science [5], data sharing and reuse [6]. For example, iDARK project (*https://www.esciencecenter.nl/project/idark*), which aims to collect existing dark matter models, will be using Spot in order to allow researchers to compare the particle physics data sets (see *http://www.idarksurvey.com*).

This abstract gives an overview of main features of Spot, an open source data analytics platform, and its capabilities.

Software Components

The software builds on a number of concepts from the field of information visualization: It uses multiple coordinated views to show the data from different perspectives. All charts allow direct manipulation (i.e., selecting and zooming) of the data, and provide visual clues or animations when data changes due to the user interaction.

Spot consists of three components: a *framework*, a *frontend*, and a *server*. A brief description for each of these components are given in the following subsections.

Spot-framework: Spot-framework provides classes for data sets, data views, partitions, aggregation and filtering. A *data set* consists of a number of items (or rows), and each item has a set of *facets*. Facets can be used to partition the data, or they can be aggregated (counted). For numerical data more complex operations are possible, namely, summed, averaged, extremes, standard deviation. One or more facets make up a *filter*, and all filters combined together form the *data view*. All filters in a data view are linked, and a change in one filter triggers a update of the whole data view.

Frontend: The frontend is a web-based application. It has separate pages where a user can upload and define data sets, a dashboard page that provides the main interaction, and a page where analysis can be downloaded or shared. The frontend is built on a number of open source JavaScript packages, such as Chart.js, Vis.js, and Sigma.js for animated plots, and Ampersand.js and Material Design Lite for the interface. The visualization libraries are HTML Canvas based, and offer OpenGL accelerated rendering where available.

Spot-server: Spot-server processes requests for data and applies the necessary filtering and aggregation. When the data is available, it is pushed to the client which can then update the charts.

Spot currently has two different implementations for the server component. The first one, which is included in Spot-framework, is based on Crossfilter.js and runs in the user's web browser without requiring any further resources or even internet access. The second implementation provides a bridge to an external PostgreSQL database for scalability. Database queries are run in parallel, and make use of indices for extra performance.

Software Functionalities

Data import and database connection: There are two options to import a new data set. In the first option, users can upload data available on their own system. In order to make the data import process easy for many scientific domains, most common data formats (e.g. CSV and JSON) are supported. After the import, they are checked to automatically detect data types, such as integers, date and strings. Users can then fix auto detection issues. In the second option, the data is imported from Spot-server. The meta data for a data set (e.g. the name and the description) can easily be set in a configuration file stored on server-side.

Dashboard: Charts are added to the dashboard by clicking the chart icon. A chart requires one or more facets to partition over, and can take up to 4 facets to aggregate. Charts show their configuration pane by default. A visual feature of the chart can be linked to a specific facet by dragging a facet from the top of the screen and dropping it on a slot in the configuration pane. A partition or aggregation can be further configured by clicking on its name on the configuration pane.

Download and share: The generated dashboard can be saved as a single file, a session file, in JSON format. This file contains aggregated data and dashboard settings such as existing charts, existing filters. The session file then can be used to restore the analysis. Moreover, it can also be uploaded to a cloud storage and a link to the session can be shared.

ACKNOWLEDGMENT

This work is supported by the Netherlands eScience Center under the project iDark: The intelligent Dark Matter Survey.

REFERENCES

[1] Plotly. Dash, 2018.
[2] Christopher Ahlberg. Spotfire: an information exploration environment. *ACM SIGMOD Record*, 25(4):25–29, 1996.
[3] Tableau Foundation. The Tableau Platform, 2008.
[4] Jisk Attema and Faruk Diblen. Nlesc/spot: Version 0.1.0, 2017.
[5] Jorge Machado. Open data and open science. *Open Science*, page 189, 2015.
[6] Mark D Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E Bourne, et al. The fair guiding principles for scientific data management and stewardship. *Scientific data*, 3, 2016.

*This work is supported by the Netherlands eScience Center under the project iDark: The intelligent Dark Matter Survey.

[1]Netherlands eScience Center, Science Park 140, 1098 XG Amsterdam, The Netherlands f.diblen@esciencecenter.nl

[2]Institute for Mathematics, Astro- and Particle Physics IMAPP, Radboud Universiteit, Nijmegen, The Netherlands

978-1-5386-9156-4/18/$31.00 ©2018 IEEE
DOI 10.1109/eScience.2018.00127