

CrossWidget: a D3.js plugin to support multiple scented cross filtering activities

G. Blasilli¹ , S. Lenti¹  and A. Palleschi¹ 

¹Sapienza University of Rome, Italy

Abstract

Filtering is one of the basic interaction techniques in information visualization. To better support a filtering activity, the visualization should be enriched by selectors composed of different visual elements and data distribution (i.e., scented widgets), in order to improve the navigation of information spaces and the comprehension of the filtering result. Thus, we propose CrossWidget, an open source JavaScript D3.js plugin, aimed at supporting cross filtering activities. To better perform multiple selections, to understand their result, and to better investigate the relationships among attributes of the dataset, this plugin proposes a set of “multiple scented widgets” that can be easily customized and integrated into Web visualization systems.

CCS Concepts

• **Human-centered computing** → Visualization systems and tools; Visualization toolkits;

1. Introduction

Filtering activity is one of the basic interaction techniques in information visualization, and it has been an important research field for many years. In 1994, Shneiderman proposed dynamic queries [Shn94] aimed at supporting the user in querying by providing continuous feedback. In order to support this activity, visualizations are very often enriched with specific components (e.g., sliders, checkboxes, radio buttons) that strongly depend on the data type (see, e.g., attribute explorer [ST98] of Spence and Tweedie). These components are usually improved by combining them with information on the data distribution, as proposed by Eick [Eic94] that superimposes the data distribution on a slider or Wittenburg *et al.* [WLHS01] that provide the feedback on data distribution using bargrams. While the most common and adaptable filtering components are probably the range sliders [LN03], several innovative approaches exist, such as the interactive legend, proposed in [RLP10], which is simultaneously legend and filter. It is common to apply to the data a sequence of selections instead of just one and some works specifically face this problem (e.g., in the work of Zhao *et al.* [BCZP11] the user can investigate time-data series with a “lens” or a sequence of them; in the work of Heer *et al.* [HAW08] a sequence of selections can be applied generating different visualizations). Furthermore, the analysis of multivariate data has specific characteristics; in particular, the need to highlight relationships across dimensions that have different domains whose require different visual abstractions as presented by Weaver in [Wea10].

We decided to create *CrossWidget*, a D3.js [BOH11] plugin, to support the creation of the filtering components for multivariate dataset, specifically focusing on the notion of *scented widget* pre-

sented by Willet *et al.* in [WHA07], in which a widget (e.g., slider) and a visual scent (e.g., boxplot) are combined together to provide information scent cues for navigating information spaces. The plugin allows to easily generate multiple scented widgets by combining several widgets and visual scents that can be easily customized and integrated in Web visualization system to better investigate the relationships among dimensions of the dataset [SC16]. More in details, for every dimension of the dataset it is possible to setup: one or more widgets that allow to filter items, possibly integrating further feedback about the selection in order to maintain the context of cross filtering activities, and one or more visual scents that summarize properties of the entire dataset and of the current selection.

Source code and a live demonstration of *CrossWidget* are available at <https://github.com/aware-diag-sapienza/crosswidget>.

2. CrossWidget

We implemented *CrossWidget* as an open source JavaScript plugin of D3.js, based on Crossfilter [Cro], a library for exploring large multivariate datasets. Our library aims for visually support filtering and exploration of large multivariate datasets, by providing *multiple scented widgets* that can be easily integrated into Web visualization systems. The plugin offers a set of visual scents and a set of widgets that can be freely combined by the user to create a multiple scented widget, related to an attribute of the dataset. Widgets and visual scents are placed one on top of the other, vertically aligning their domains, in order to be comparable.

Visual Scents. They are designed to show information of a single attribute concerning all the dataset items and/or only a selection of them, with a continuous or discrete representation. The user can set the visual scent to show information about both dataset and selection or just one of them. When the current selection changes, the parts of the visual scent encoding the selection are automatically updated in order to show the new selected items. By default, the domain of a visual scent is set equal to the domain of the dataset. Since the domain of the selection can be much smaller than the domain of the entire dataset, a visual scent that shows only the selection could be difficult to understand. To avoid that, the user can decide to enlarge the visualization of the selection to fit the whole size of the scent, improving its readability. We have currently implemented the following visual scents:

- **Boxplot.** This scent shows the distribution of the entire dataset and/or the selection using two vertically aligned boxplots. This visual scent cannot be used with categorical data.
- **Frequency plot.** This scent shows the distribution of the entire dataset and/or the selection using a stacked histogram representing a frequency plot.
- **Heatmap.** This scent shows the distribution of the entire dataset and/or the selection using two vertically aligned one-dimensional heatmaps. The domain of the visual scent is divided in a custom number of rectangular bins in which the background color encodes the number of items in the bin.
- **Violin plot.** This scent shows the estimation of the probability density function of the entire dataset and/or the selection using a stacked area chart. The density function is estimated using the Kernel Density Estimation (KDE) method [Ros56].

Widgets. These elements are designed to allow filtering activities on attributes of the dataset. Since multiple widgets can be used for an attribute, and on each widget multiple selections are allowed, the query on the attribute is defined as the logical *OR* among all the selected intervals of the attribute. If no selection is done over an attribute, all the domain is considered selected. The resulting selection on the dataset is defined as the logical *AND* among all the attribute selections. We have currently implemented the following widgets:

- **Continuous selector.** This widget represents a typical selector used to perform queries on a continuous domain as a range slider [AS94]. This widget has an area representing the selected interval of the attribute that can be moved on any two points of the domain defining the selection interval.
- **Fixed intervals selector.** This widget allows the user to select, like a checkbox, a number of predefined no-overlapping intervals of the domain. Intervals can be either user defined or equally spaced in the domain. The selection of intervals is done by clicking on their area. On this widget, if enabled by the user, the area of an interval can encode the *selection feedback*, i.e., the amount of items in the interval that are currently selected: an empty area means that no items of the interval are currently selected; a striped area means that some of the items are currently selected; a fulfilled area means that all the items are currently selected.

Plugin Usage. A crosswidget needs a crossfilter object and an accessor function to be linked to a dimension, and it is composed of several widgets and/or visual scents, placed one on top of the other. Figure 1 shows some crosswidgets applied to the Iris dataset; the following code illustrates how to create the top left one (sepal length), containing a boxplot scent and a slider selector:

```
// the crossfilter object
var iris = crossfilter(...);
// create a crosswidget on sepal length
var cwSepalLength = d3.crosswidget(iris, function
(d) { return d["sepal-length"]; })
  .addComponent(
    d3.crosswidgetComponent.boxplot()
  )
  .addComponent(
    d3.crosswidgetComponent.sliderSelector()
  );
// draw the crosswidget into a div
d3.select("#sep-len-div").call(cwSepalLength);
```

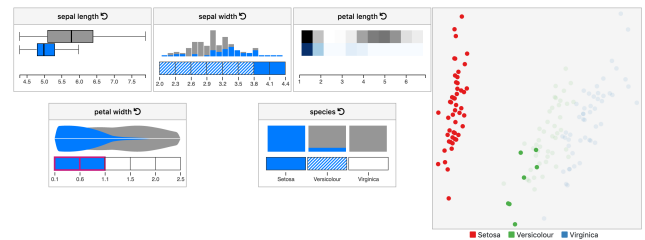


Figure 1: A toy visualization system of the Iris dataset containing five crosswidgets and a PCA scatterplot. The current selection (blue) is composed by iris having petal width < 1.1, and it is compared to the entire dataset distribution (grey). The selection feedback on species shows that: all Iris Setosa are included (full filled blue area); some Versicolour iris are selected (striped blue area); no Virginica flowers appear in the selection (white area).

3. Conclusions and Future Work

We have presented *CrossWidget* a D3.js plugin that generates multiple scented widgets to be easily integrated into Web visualization systems, supporting cross filtering activities on multivariate datasets. Multiple scented widgets are developed in order to better understand the result of a sequence of selections, allowing to investigate relationships between different attributes of the dataset.

Concerning scents and widgets based on bins (e.g., frequency plot, fixed interval selector) a current limitation is the number of bins that is defined only at creation time; therefore we are working on a way to dynamically change binning. Furthermore, we are investigating on how to better represents an high number of bins in a limited space, in order to maintain high the readability of the provided information. Inspired by Selection Mondrian [The02], we are working on developing additional scents and widgets (e.g., ridge-line plots), also allowing the user to change logical operators between widgets. Eventually, we plan to evaluate the multiple scented widgets with respect to specific tasks, in order to suggest to the user the most useful combination for its purposes.

References

- [AS94] AHLBERG C., SHNEIDERMAN B.: Visual information seeking: Tight coupling of dynamic query filters with starfield displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 1994), CHI '94, ACM, pp. 313–317. doi:10.1145/191666.191775. 2
- [BCZP11] BALAKRISHNAN R., CHEVALIER F., ZHAO J., PIETRIGA E.: Exploratory analysis of time-series with chronolenses. *IEEE Transactions on Visualization & Computer Graphics* 17, 12 (dec 2011), 2422–2431. doi:10.1109/TVCG.2011.195. 1
- [BOH11] BOSTOCK M., OGIEVETSKY V., HEER J.: D³ Data-Driven Documents. *IEEE Transactions on Visualization and Computer Graphics* 17, 12 (dec 2011), 2301–2309. doi:10.1109/TVCG.2011.185. 1
- [Cro] Crossfilter. URL: <https://github.com/crossfilter/crossfilter>. 1
- [Eic94] EICK S. G.: Data Visualization Sliders. *Proceedings of the 7th ACM Symposium on User Interface Software and Technology - UIST '94* (1994), 119–120. 1
- [HAW08] HEER J., AGRAWALA M., WILLETT W.: Generalized selection via interactive query relaxation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2008), CHI '08, ACM, pp. 959–968. doi:10.1145/1357054.1357203. 1
- [LN03] LI Q., NORTH C.: Empirical comparison of dynamic query sliders and brushing histograms. In *IEEE Symposium on Information Visualization 2003 (IEEE Cat. No.03TH8714)* (Oct 2003), pp. 147–153. doi:10.1109/INFVIS.2003.1249020. 1
- [RLP10] RICHE N. H., LEE B., PLAISANT C.: Understanding interactive legends: a comparative evaluation with standard widgets. *Computer Graphics Forum* 29, 3 (2010), 1193–1202. doi:10.1111/j.1467-8659.2009.01678.x. 1
- [Ros56] ROSENBLATT M.: Remarks on some nonparametric estimates of a density function. *Ann. Math. Statist.* 27, 3 (09 1956), 832–837. doi:10.1214/aoms/1177728190. 2
- [SC16] SUN M., CONVERTINO G.: Intervert: Drilling into categorical-numerical relationships. In *Proceedings of the International Working Conference on Advanced Visual Interfaces* (New York, NY, USA, 2016), AVI '16, ACM, pp. 44–51. doi:10.1145/2909132.2909253. 1
- [Shn94] SHNEIDERMAN B.: Dynamic queries for visual information seeking. *IEEE software* 11, 6 (1994), 70–77. 1
- [ST98] SPENCE R., TWEEDIE L.: The attribute explorer: information synthesis via exploration. *Interacting with Computers* 11, 2 (1998), 137–146. 1
- [The02] THEUS M.: Interactive data visualization using mondrian. *Journal of Statistical Software* 7, 11 (11 2002), 1–9. URL: <http://www.jstatsoft.org/v07/i11>. 2
- [Wea10] WEAVER C.: Cross-filtered views for multidimensional visual analysis. *IEEE Transactions on Visualization and Computer Graphics* 16, 2 (March 2010), 192–204. doi:10.1109/TVCG.2009.94. 1
- [WHA07] WILLETT W., HEER J., AGRAWALA M.: Scented widgets: Improving navigation cues with embedded visualizations. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (Nov 2007), 1129–1136. doi:10.1109/TVCG.2007.70589. 1
- [WLHS01] WITTENBURG K., LANNING T., HEINRICHS M., STANTON M.: Parallel bargrams for consumer-based information exploration and choice. In *Proceedings of the 14th annual ACM symposium on User interface software and technology* (2001), ACM, pp. 51–60. 1