CrossWidgets: Enhancing Complex Data Selections through Modular Multi Attribute Selectors

Conference Paper · September 2020

DOI: 10.1145/3399715.3399918

5 authors, including:

Marco Angelini
Sapienza University of Rome
81 PUBLICATIONS **757** CITATIONS
SEE PROFILE

Graziano Blasilli
Sapienza University of Rome
15 PUBLICATIONS **95** CITATIONS
SEE PROFILE

Simone Lenti
Sapienza University of Rome
20 PUBLICATIONS **155** CITATIONS
SEE PROFILE

Alessia Palleschi
Sapienza University of Rome
9 PUBLICATIONS **41** CITATIONS
SEE PROFILE

CrossWidgets: Enhancing Complex Data Selections through Modular Multi Attribute Selectors

Marco Angelini
Sapienza University of Rome
angelini@diag.uniroma1.it

Graziano Blasilli
Sapienza University of Rome
blasilli@diag.uniroma1.it

Simone Lenti
Sapienza University of Rome
lenti@diag.uniroma1.it

Alessia Palleschi
Sapienza University of Rome
palleschi@diag.uniroma1.it

Giuseppe Santucci
Sapienza University of Rome
santucci@diag.uniroma1.it

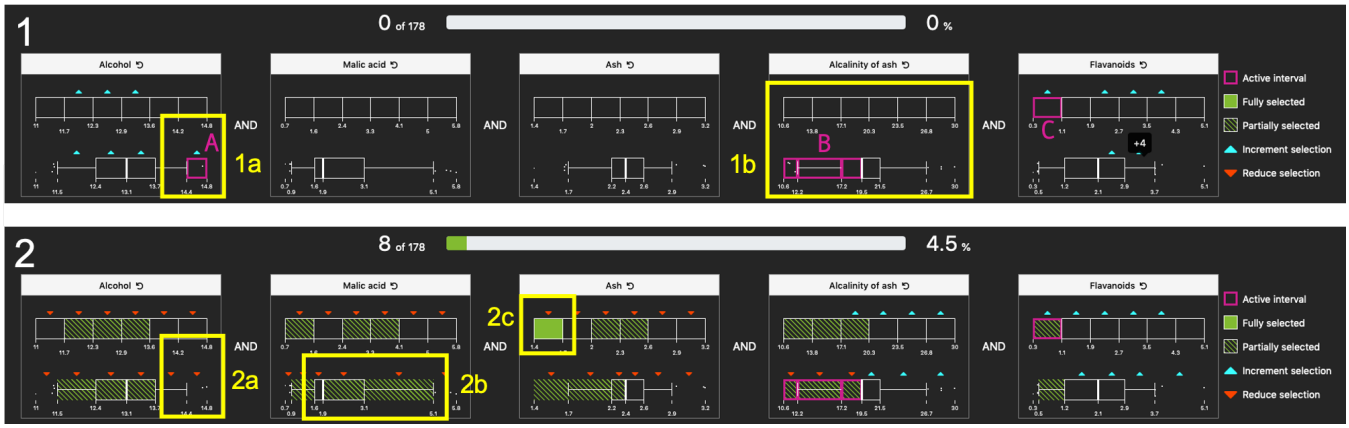

Figure 1: Exploring the Wine dataset using the proposed *CrossWidgets*. 1) The user looks for wines with the highest values of Alcohol (*A*), Alcalinity of ash below the median (*B*), and Flavanoids in the first sixth of the domain (*C*), getting an empty set of wines. The guidance (light blue arrows) suggests how to relax the imposed constraints in order to increase the selection. It is useless to change the condition on Alcalinity of ash because it would not change the selection (*1b*), and the same holds for Malic acid and Ash, while guidance suggests to relax the condition on the alcohol outliers with highest value (*1a*). 2) Using the guidance (*2a*), the number of selected items increases. The set based feedback (the green filling of the selectors intervals) suggests the relationship of the selected 8 wines with the displayed attributes: in particular, all the wines (solid green) with a value of Ash in the first sixth of the domain are in the selection (*2c*) and the selection partially includes wines with Malic acid between median and third quartile (*2b*).

ABSTRACT

Filtering is one of the basic interaction techniques in Information Visualization, with the main objective of limiting the amount of displayed information using constraints on attribute values. Research focused on direct manipulation selection means or on simple interactors like sliders or check-boxes: while the interaction with a single attribute is, in principle, straightforward, getting an understanding of the relationship between multiple attribute constraints and the actual selection might be a complex task. To cope with this problem, usually referred as cross-filtering, the paper provides a general definition of the structure of a filter, based on domain values and data distribution, the identification of visual feedbacks on the relationship between filters status and the current selection, and guidance means to help in fulfilling the requested selection. Then, leveraging on the definition of these design elements, the paper proposes CrossWidgets, modular attribute selectors that provide the user with feedback and guidance during complex interaction with multiple attributes. An initial controlled experiment demonstrates the benefits that CrossWidgets provide to cross-filtering activities.

KEYWORDS

visual filtering; crossfilter; visual guidance; user feedback

ACM Reference Format:
Marco Angelini, Graziano Blasilli, Simone Lenti, Alessia Palleschi, and Giuseppe Santucci. 2020. CrossWidgets: Enhancing Complex Data Selections through Modular Multi Attribute Selectors. In *International Conference on Advanced Visual Interfaces (AVI '20), September 28-October 2, 2020, Salerno, Italy*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3399715.3399918

AVI '20, September 28-October 2, 2020, Salerno, Italy

ACM ISBN 978-1-4503-7535-1/20/09…$15.00
https://doi.org/10.1145/3399715.3399918

1 INTRODUCTION

Filtering is a common activity in Information Visualization, and it has been around from such a long time that it is commonly considered a stable and understood practice. Also, the challenging issue of providing users with filtering feedback while interacting with suitable selectors (e.g., sliders) dates back, at least, to 90s, see, e.g., the pioneering Ben Shneiderman's dynamic query idea [20] and Spence and Tweedie attribute explorer [25]. From that time on, several proposals have dealt with solutions able to inform the user on dataset characteristics useful for *filtering activities*. In this paper we use the term *Filtering Overview* to denote techniques and feedback explicitly intended for helping the user's filtering activities, to distinguish it by the related *data overview*, that refers to the means used for providing an overview of the dataset (see, e.g., [21]). *Filtering Overview* has been investigated in several contexts by exploiting different characteristics of the whole dataset and of the current selection (i.e., the data subset corresponding to the actual selectors state) like:

Attribute domain. The most used selector for filtering is the slider. It provides the user with information about the range of values of an attribute and allows for selecting one or more intervals of values that filtered objects must satisfy. Other solutions rely on a predefined set of intervals, or for direct input of numerical values. Other techniques rely on direct manipulation, e.g., allowing to filter data on a scatter plot or on a parallel coordinates plot through brushing, or, in general, on any visualization that makes explicit and manipulable the domain of one or more attributes. In summary, regardless of the interaction technique and the used selectors, the user is aware of the attribute(s) domain(s) and can select one or more filtering intervals.

Data distribution. Other solutions exploit the explicit usage of data distribution, like superimposing the data distribution on a slider presented by Eick in [7], or the bargrams, formally introduced by Wittenburg *et al.* in [28] that provide both feedback on data distribution and selection means. Visualizing data distribution obviously implies the explicit presence of the domain values range and, according to that, it can be considered as using *both* domain and distribution.

Guidance. Some filtering solutions provide either orienting or directing guidance, for profitably interacting with multiple selectors, in order to control the number of elements in the current selection. This idea raises from the empty query paradox: the user selects several filtering intervals along with multiple attributes and the answer, obtained by logical AND, is empty. Guidance points out the conditions the user has to relax, i.e., which attribute intervals must be extended to add elements to the current selection. Some proposed techniques rely on the idea of marking dataset elements with a color that denotes that relaxing just one or two conditions will make these elements appear in the current selection (see, e.g., [25, 28]). This approach implies to duplicate all the elements for each selector and it is likely to fail when dealing with large datasets, in which it is not possible to guarantee the clear visibility of each element. Moreover, it uses a large amount of space for each selector.

As a last consideration, while the interaction with a single attribute is, in principle, straightforward, getting an understanding of the relationship between attribute constraints and the actual selection is a complex task. This complexity holds even if dealing with few attributes and limiting the underlying logic to the usual solution that uses OR combinations within an attribute with AND combination among multiple attributes (conjunct of disjuncts), see, e.g., [21].

Most of the proposals mentioned above have been published about twenty years ago and, while the principles behind them are still valid and agreed, the practical impact on filtering activities is not so evident. We are still witnessing many Information Visualization solutions that delegate to a list of simple sliders the burden of filtering the dataset while focusing on valuable visualizations and direct manipulation interactions. Likely, this is due to the extra space the filtering components solutions require, limiting the room for the main visualizations used in Infovis solutions.

This paper tries to resume these old lines of thought introducing and formalizing an integrated set of techniques, called *CrossWidgets*, consisting on modular user interface elements containing attribute selectors, visual feedback and guidance, useful for supporting the implementation of a *Filtering Overview*. The goal of *CrossWidgets* is to provide the user with all the elements that have been tickled in the previous paragraphs *at selectors level*, abstracting from both the visualization of the dataset and the direct manipulation techniques. In particular, the proposed approach relies on both attribute domain and data distribution. Using different selectors it provides a guidance that is independent from data visualization and a *set based feedback* that facilitates the user comprehension of the relationship that exists between the current selection and the different constraints that have been set through multiple selectors. The paper proposal has been evaluated with a controlled user study involving 28 users, comparing it with the traditional sliders approach and testing separately, the advantages of selectors with (a) domain and distribution and (b) domain and distribution plus guidance and set based feedback. Summarizing, the contributions of the paper are the following:

- the definition and the formal characterization of the main aspects that play a relevant role in providing an effective *Filtering Overview*;
- the design of the *CrossWidgets* i.e., modular user interface components including selectors, visual feedback and guidance. The composition of one or multiple *CrossWidgets* effectively support the *Filtering Overview* concept;
- implementation of an initial solution based on *CrossWidgets*;
- the outcomes of a controlled experiment involving 28 users and aiming at getting insights on the advantages and drawbacks of the proposed approach.

The paper is structured as follows: Section 2 describes related proposals. Section 3 defines and characterizes all the elements we use for implementing the *Filtering Overview* through the *CrossWidgets*, visual components that include selectors, visual scents and user feedback. The structure of a *CrossWidget* is described in Section 4. Section 5 presents the controlled experiment and the main hypothesis we have derived from it, and Section 6 concludes the paper.

2 RELATED WORK

Conducting a general data analysis process usually requires the analyst to interact with the system, and interaction always plays a central role in Visual Analytics and Information Visualization. Yi *et al.* [29] present a deep study about the significance of interaction in all the operations that a visual system provides: they identify filtering and reviewing filtering techniques and highlight the effort needed to visually and interactively support the filtering activities. Among them, the main ones are dynamic queries support (see, e.g., [23]), attribute exploration (see, e.g., [25]), and direct manipulation of the visualization (see, e.g. [9]). However, no efforts are mentioned about proposing more informative and guiding means for better supporting the user in filtering operations: the use of a generic form of selectors like sliders or buttons, and simple implementation of guidance, typically rely on the inspection of dataset elements (see, e.g., [28]).

To the best of the authors' knowledge, previous works in visualization community focused more on direct manipulation of visual elements in order to support filtering than relying on visual cues on selectors themselves. Scalability considerations point out the advantages of the second solution, where the more dimensions the visualized dataset has the more visual abstraction is needed in order to govern the data exploration. Nonetheless there exist cases in which dimensions require to be individually analyzed and their domain and distribution can be fruitfully used to filter the whole dataset (e.g., filtering data on specific user requirements, or dealing with the most relevant dimensions identified by a Principal Component Analysis, see, e.g., [13]). The best examples for the latter case are the different implementations of interactive brushing histograms (see, e.g., [26]) allowing fast multidimensional filtering on a dataset through different variants of visual means; among them, the most used are range sliders for each attribute [16].

In all these cases, providing a *Filtering Overview* is crucial to make the user aware of the actual state of the analysis process and where she can proceed in order to fulfill her information needs. Besides the pioneering works cited in the introduction, initial efforts in proposing more informative interaction selectors for data visualization can be found in the paper from Eick [7], where data filtering sliders are proposed, coordinating a slider with the corresponding data distribution. Ahlberg and Shneiderman [1] propose a dynamic queries environment embedding simple feedback on the sliders for guiding the user, i.e., making explicit in which direction the slider must be moved for reducing the cardinality of the result. Willet *et al.* [27] propose scented widgets, user interface elements enriched with embedded visualizations that facilitate navigation in information spaces. Differently from our approach, they consider the filtering elements only in direct connection to the visualization and do not explore the mutual relations that can exist among two or more scented widgets nor they target the support of a general *Filtering Overview*.

Literature on relationships among sets comprehends several efforts: Gratzl *et al.* [8] propose a solution for visualizing relationships among sets coming from different tabular datasets, while Rodgers *et al.* in [17] present a survey on Euler diagrams. More recently, Simonetto *et al.* [22] propose a solution for better representing relationships among sets using the Euler diagrams. Differently from our approach the authors do not consider the problem of representing the effect of possible filtering activities directly on selectors and to relate set relationships to selector data intervals. Lex *et al.* [15] propose an interesting system for studying behavior of intersecting sets, analyzing relations among data: the work is aimed at providing deeper comprehension of the existing relationships among sets through a complete visual system, while it does not cope with visual enrichment of selectors for guiding the user as our proposed solution envisions. Regarding guidance, existing literature coped with the problem of providing insights or recommendations for the next step to execute in a data analysis process. Several works focused on high-level aspects, modeling and characterizing the problem in Information Visualization, see, e.g., Schulz *et al.* [19] and Visual Analytics, see, e.g., Ceneda *et al.* [5]. Other proposals focused on low-level aspects, like suggesting areas of interaction in the visualization, see, e.g., Boy *et al.*, in [4] and/or best data representation, see, e.g., Behrisch *et al.* [3]. Sarvghad *et al.* [18] and Xia *et al.* [12] propose a way to assess which dimensions are the most used in a multidimensional dataset analysis, and which are the relationships among them. In particular, both approaches propose a novel and/or additional visualizations for communicating results, differently from our work that focuses on selectors informativeness.

Overall, while filtering remains a central aspect studied in visualization research, less efforts are spent toward providing what we define as *Filtering Overview*, for which a characterization is provided in the next section.

3 FILTERING OVERVIEW

The visual exploration of a dataset, according to the well-known Visual Information Seeking Mantra proposed by Shneiderman [21], usually involves a three-step process: overview first, zoom and filter, then details-on-demand. The goal of the filtering step is to select an interesting subset of the dataset by filtering out unwanted entries. As stated by Keim [14], this step is usually accomplished following two main strategies: selection of the desired subset by direct manipulation (browsing) or specification of the properties of the subset (querying). This work focuses on the second strategy and in particular on the definition of a *Filtering Overview* from the selectors supporting it. The *Filtering Overview* is independent of the well-known visualization overview and is obtained through a combination of elements that govern the filtering operations, that we defined as *CrossWidgets*. Its goals are to support the user in filtering operations, to provide the user with feedback on the performed actions, in terms of properties of the current selection, and to visually guide the user for obtaining the desired selection. In principle, a user should be able to obtain the desired data selection by simply looking at the *Filtering Overview*, while referring to the rest of the visual environment for more semantically meaningful operations.

Focusing on a single *CrossWidget*, the design elements that we identified are the data attributes, the selector properties, the selector relationships and the visual scent. These design elements are modeled in the following.

Let us consider a dataset with p entries $\{e^1, e^2, \ldots, e^p\} = E$, and q quantitative or categorical attributes $\{a_1, a_2, \ldots, a_q\} = A$. The i-th entry of the dataset is defined by the q-tuple $\{v_1^i, \ldots, v_q^i)\} = v^i$,

where v_h^i denotes the value of the attribute a_h for the e^i entry. The characteristics of the domain attributes influence the type of queries that can be carried out. In the following we focus on continuous, discrete and ordinal attributes neglecting the nominal ones, assuming for every attribute a domain $dom(a_h) = [a_h^{min}, a_h^{max}]$. Except for the nominal attributes, a typical query $q_h(x, y)$ on a single attribute a_h aims to select all the entries of the dataset that have a value of a_h between two values x and y. Formally, the resulting set is a subset $E_h(x, y)$ of E, such that:

$$E_h(x, y) = \{e^i \in E : x \leq v_h^i \leq y\}$$

3.1 Selectors

There exist different methods to express queries in Information Visualization depending on the domain and characteristics of the attributes.

Continuous range selector. A typical selector used to perform this kind of query on a continuous domain is the *range slider* [1]. This selector has a track that represents the active domain of the attribute and two "slider thumbs" that can be positioned on any two points of the domain (i.e., x and y), thus defining the range of interest.

Partition based selectors. Another possible approach to perform these queries is to use selectors based on a predefined partition of the domain. Let $P_h(J)$ be a generic partition of $dom(a_h)$:

$$P_h(J) = \{x_1, x_2, \ldots, x_{n+1}\} \text{ such that } a_h^{min} = x_1 < \ldots < x_{n+1} = a_h^{max}$$

The partition divides the domain in n intervals $\{I_1, I_2, \ldots, I_n\}$, where $I_l = [x_l, x_{l+1}]$. It is possible to associate a query $q_h(I_l)$ to each interval I_l and the answer to that query will be a subset $E_h(I_l)$ of E:

$$E_h(I_l) = \{e^i \in E : x_l \leq v_h^i \leq x_{l+1}\}$$

A selector $S_h(J)$ that uses this approach shows to the user a partition of the domain and allows her to select one or more intervals. Let $A_h(J)$ be the set of all the *active* intervals of the selector (i.e., the intervals selected by the user), the subset returned by the selector will be:

$$E_h(J) = \bigcup_{I_l \in A_h(J)} E_h(I_l)$$

For attributes with a continuous domain, the active domain is typically partitioned in uniform intervals, while discrete and categorical domains are already inherently partitioned. If the number of intervals is small enough, the user can directly select the intervals by clicking them. Conversely, when the number of intervals is higher, it is possible to select one or more intervals through a range slider on the partition eventually snapped to the intervals. Notice that the second approach is not feasible for nominal domains that do not have an inherent ordering; for this reason we exclude them from our analysis.

3.2 Relation among selectors

While the operation of a single selector is straightforward, their coordination opens some questions. By definition, the intervals of a single selector are disjointed; therefore the subset obtained by the activation of more than one interval is equal to the union of their subsets. Consider two selectors $S_h(M)$ and $S_h(N)$ on the same attribute a_h. The subsets of entries obtained by the two selectors can be partially overlapping, thus the desired selection can be both the intersection and the union of them. Both the strategies have possible application scenarios, while the most common choice (see Shneiderman [21]) is to combine selections within an attribute using a logical OR. Therefore, the resulting selection E_h obtained on an attribute a_h is:

$$E_h = E_h(M) \cup E_h(N)$$

The further step is the combination of selectors on different attributes. The typical combination of queries on different attributes uses a logical AND (see [21]); the resulting selection will be:

$$E_S = \bigcap_{h=1}^{q} E_h$$

3.3 Visual Scents

Depending on the available space and on the requirements, different ways to represent the output of the queries on the selectors were proposed. In the following, we present the aspects on which these solutions focus.

Data representation. The first element that differentiates the selectors is the representation of the data elements. In particular we identify at high level three possible representations:

- **No representation**: the data elements are not represented and the active domain is the only information reported on the selector;
- **Distribution**: a typical way to represent the data elements is to show their distribution. The encoding of the distribution depends on the representation of the domain, in particular:
  - Density plots are used for continuous domain representations;
  - Histograms (or bar charts) are typically used to encode the cardinality of the entries in discrete intervals of the domain;
- **Direct representation**: a direct representation of the elements on the selectors domain is rarely used because it does not scale.

Relationship between the current selection and selectors. For each interval I_l of the domain of a selector (or as a whole if it is continuous) it is possible to indicate the relationship $R_h(I_l, S)$ between the data elements in the interval $E_h(I_l)$ and the data elements currently selected E_S:

$$R_h(I_l, S) = |E_h(I_l) \cap E_S|$$

The value of $R_h(I_l, S)$ can be conveyed at different levels of granularity, both directly and through the percentage (i.e., $PR_h(I_l, S) = R_h(I_l, S)/|E_h(I_l)|$), or using three levels of discretization that associates to each interval I_l of an attribute a_h one of these three states:

(1) *Not Selected*, the current selection does not contain entries that have a value of a_h in the interval (i.e., $PR_h(I_l, S) = 0$);
(2) *Partially Selected*, the current selection contains some of the entries that have a value of a_h in the interval (i.e., $0 < PR_h(I_l, S) < 1$);
(3) *Fully Selected*, the current selection contains all the entries that have a value of a_h in the interval (i.e., $PR_h(I_l, S) = 1$).

where state 2 can be decomposed in predefined percentage levels like the usual box-plot quartiles (e.g., 25%, 50%, and 75%), giving rise to a thinner discretization.

Guidance for next step. A further possible enrichment recovers the concept of guidance formalized by Schulz *et al.* [19] for Information Visualization and by Ceneda *et al.* [5] for Visual Analytics. With respect to the taxonomy proposed in [5] regarding the degrees of guidance, we identify these possibilities:

Orienting guidance. Every interval can be characterized considering the number of violated constraints by the entries in the interval that do not belong to the current selection.

$$O_h(I_l, S, t) = |\text{entries in } I_l \text{ that violate } t \text{ constraints}|$$

Similarly to the considerations on the relation between the current selection and an attribute selector, this information can be aggregated at different levels of granularity. Tweedie *et al.* [25] formalize it as *sensitivity* providing this information at 4 levels (one, two, three, and more than three violations).

Directing guidance. Given the current selection E_S, let $E'_S(I_l)$ be the selection obtainable by triggering the state of the interval I_l.

$$D_h(I_l, S) = \begin{cases} |E'_S(I_l) \setminus E_S|, & \text{if } |E'_S(I_l)| \geq |E_S| \\ -|E_S \setminus E'_S(I_l)|, & \text{otherwise} \end{cases}$$

The value of $D_h(I_l, S)$ clearly indicates if and how the triggering of the interval will change the selection; again, it is possible to explicitly provide the value or to aggregate it up to a three level discretization that associates to each interval I_l of an attribute a_h one of these three states:

(1) *No variation*, triggering the interval does not modify the current selection (i.e., $D_h(I_l, S) = 0$);
(2) *Increment selection*, triggering the interval increases the number of entries in the current selection (i.e., $D_h(I_l, S) > 0$);
(3) *Reduce selection*, triggering the interval decreases the number of entries in the current selection (i.e., $D_h(I_l, S) < 0$).

Also in this case, state 2 and 3 can be decomposed in predefined percentage levels like the usual boxplot quartiles (i.e., 25%, 50%, and 75%), giving rise to a thinner discretization.

Prescribing guidance. To the best of the authors knowledge, there no exist examples of prescribing guidance on the selectors.

4 CROSSWIDGETS

According to the notion of *scented widget* presented by Willet *et al.* [27], 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, we have defined the term *CrossWidgets* as a modular user interface element implementing the formalization defined in Section 3. A *CrossWidget* is a modular user interface element, linked to a specific attribute of the dataset, containing multiple selectors and visual scents that provide the user with feedback and guidance. The presence of multiple *CrossWidgets* on different attributes of the dataset helps and guides the user during complex interaction with multiple attributes. As a proof of concept, we have implemented and used in a pilot test *CrossWidgets* with a filtering mechanism based on the combination of two selectors for each attribute (see Figure 2). The first one is thought to make selections based on the domain of the attribute, the second one on the distribution of the data regardless their cardinality. The *domain selector* divides the domain in n intervals of equal width, independently from the distribution of the values:

$$P_h(dom) = \{x_1, x_2, \ldots, x_{n+1}\} \text{ such that } x_{l+1} - x_l = \frac{|dom(a_h|}{n}$$

The aim of the second selector is to provide significant intervals of the data distribution. Among the several possible choices, we have decided to use the quartiles representation presented by Tukey [24]. The *box-plot selector* divides the domain in 6 intervals according to that representation. Data are depicted in groups through their quartiles, in particular the underlying partition $P_h(box)$ is equal to $\{a_h^{min}, p_2, q_1, q_2, q_3, p_{98}, a_h^{max}\}$ where p_k is the k^{th} percentile, and q_k is the k^{th} quartile. Selectors are positioned on top of each other in order to align their domains. The selection occurs by clicking on a interval; the color of the border (pink) of an interval encodes if it is selected. The area of an interval encodes the amount of entries in the interval currently selected:

- Empty area: no entries of the interval are currently selected;
- Striped area: some of the entries are currently selected;
- Filled area: all the entries are currently selected.

This type of implemented *CrossWidget* provides also a directing guidance by means of arrows aligned on top of the intervals. The absence of an arrow indicates that the triggering of the interval does not change the selection, the presence of a blue or a red arrow indicates an increment or decrement of the selection, respectively.

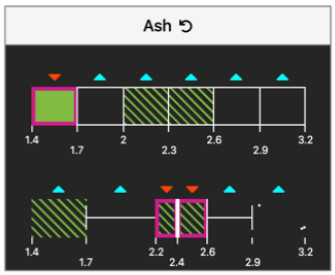

Figure 2: A *CrossWidget* proof of concept on the Ash attribute of the Wine dataset, using a Partition based selector and a boxplot. Purple rectangles show the user selection and set based feedback is available: not selected (black), partially selected (shaded green), and fully selected states (solid green) are encoded on each domain interval. Light blue and red arrows encode guidance for increasing and decreasing the result cardinality, respectively.

5 EVALUATION

We have conducted a formal user study, i.e., a controlled experiment, to evaluate effectiveness (i.e., the accuracy with which the user achieves specific goals, see [10]) and efficiency (i.e., the effort in relation to effectiveness, see [10]) of different configurations of the *CrossWidgets* proposed in Section 4. The experiment involved

Figure 3: The prototype showing a selection of 43 elements and one active interval on the domain-based selector: *Malic Acid* [0.7, 1.6]. The scatterplot shows all elements of the dataset, highlighting in yellow the selected ones. The number of total and selected elements is reported below the scatterplot. Different *CrossWidgets* are shown at the bottom. Each of them contains a domain selector and a box plot selector for the same attribute. Set-based feedback and guidance are visible on the intervals.

28 computer science graduated people (22 males, 6 females). Participants have been asked to answer 8 questions and we have collected their answers, tracing answering times and interactions with selectors.

5.1 Validation Prototype

We have implemented a validation prototype, dividing its interface into two parts: a) a scatter plot chart on the top and b) a selector panel on the bottom containing the *CrossWidgets* (see Figure 3). Given a dataset and its attributes, two of them are used by the scatter plot to show all the elements of the dataset, highlighting in yellow the selected ones. The bottom panel contains, instead, the *CrossWidgets* associated to a subset of the remaining attributes and some basic information about the actual selection composition: number and percentage of selected entries.

The validation prototype can be customized by setting:

- The two attributes used in the scatter plot;
- The attributes used from the *CrossWidgets*. All the dataset attributes, or a custom subset, can be chosen (except the two attributes assigned to the scatter plot);
- Selectors included in each *CrossWidget*: an arbitrary subset of the domain, boxplot, and slider selectors;
- The presence of set based feedback;
- The presence of guidance.

In order to correctly guide the user along with the experiment execution steps, i.e., reading questions, interacting with selectors and reporting responses on a questionnaire, we have used STEIN [2], an evaluation environment that allows for quickly integrating the system under evaluation with the questions that have been designed for the user study, tracing user's activities. All user actions and elapsed times for answering the questions are stored together with the answers, allowing for a more in-depth and better evaluation of the user behavior.

The validation prototype and the demonstrative video are available at *https://aware-diag-sapienza.github.io/filtering-overview*.

5.2 Evaluation Tasks

Participants answered 8 questions grouped according to three high-level tasks. Questions are based on the *Wine*[6] dataset using four attributes: *Alcohol*, *Malic acid*, *Ash*, and *Alcalinity of ash*.

Task 1 - Cardinality selection - This task validates the *CrossWidgets* support for selecting a given number of elements.

Q1 Given the dataset and the 4 attributes, select 50% of the items.
Q2 Given the dataset and the 4 attributes, select 100 items.
Q3 Given the actual empty selection select the arbitrary modification of one of the three active attributes that selects the maximum number of wines.

Task 2 - Attributes values filtering - This task validates the *CrossWidgets* support for selecting elements which attributes satisfy some specific constraints about data distribution.

Q4 Given the actual dataset and the 4 attributes, select a not empty set of items that have 2 attributes $\geq$ median AND 2 attributes $\leq$ median.
Q5 Given the actual selection and the 25 selected wines, restrict the selection to the 12 wines with the highest *Alcohol*.

Task 3 - Current selection analysis This task validates the *CrossWidgets* support for relating the current selection to the selectors state.

Q6 Given the actual selection and the 25 selected wines, there are wines with *Alcohol* > 15?

Q7 Given the actual selection and the 25 selected wines, there are wines with *Alcohol* < 13?

Q8 Given the actual selection and the 44 selected wines, are all the wines with *Ash* > 3 included in the selection?

5.3 Methodology

In order to evaluate the model, people participating in the evaluation were divided into three groups (9, 9, 10 people), each of them assigned to a different configuration of the validation prototype:

Group 1 *CrossWidgets* use only slider selectors; no set-based feedback and no guidance.

Group 2 *CrossWidgets* use Pseudo Bargram and Box plot selectors; no set-based feedback and no guidance.

Group 3 *CrossWidgets* use all the implemented features: pseudo Bargram, Box plot selectors, set based feedback and guidance.

Moreover, we have collected dependent variables for each group: *Score* obtained on questions, elapsed *Time* and number of *Clicks*. We did not collect the number of clicks for Group 1 because clicks are not relevant while dragging a slider. Scores relate to accuracy, while time and clicks relate to efficiency.

Before starting the test, people were instructed through a live demonstration of the validation prototype on a training dataset (i.e., the car dataset [6]). After that, participants spent about 10 minutes on their given configuration - depending on the group they belong - using the training dataset, in order to get familiar with the provided features. During this time, participants were asked to accomplish some training tasks (e.g., to select elements with given constraints) discussing their choices and receiving live feedback. Then, they were presented with the evaluation environment - that relies on a different dataset (i.e., the wine dataset [6]) - and they were first asked to answer some general questions about themselves, then to answer the 8 questions using the configuration of the validation prototype of their group. At the end participants were asked to give a feedback on their experience in using the system in terms of encountered difficulties and gained benefits, as well as to comment on the overall experience.

Concerning the questions scores, we have used in *Task 1* and *Task 2* a proportional approach considering how much the given answer is far from the correct one, using the Jaccard similarity coefficient [11] when the question answer was a set of elements. Lastly, questions in *Task 3* require a true/false answer therefore we have assigned 10 to the correct ones and 0 to the others.

We remark that the two attributes used for the scatter plot were not allowed to belong to the set used in the selectors, to avoid the scatter plot might provide feedback about the filtering operations.

5.4 Results

We have collected three dependent variables: *score*, *time*, and *number of clicks* (only for Group 2 and Group 3) that are reported on Figure 4. To validate their significance we have performed a one-way between subjects ANOVA to compare the effect of different

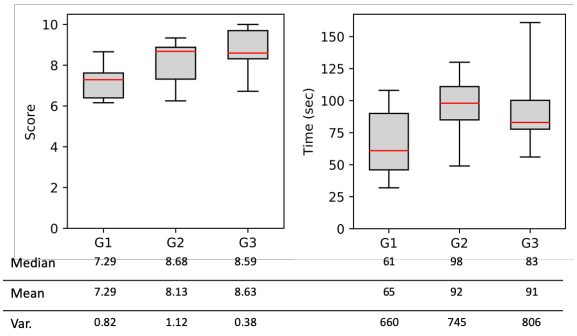

Figure 4: Distribution of the average scores obtained by participants per group, giving to each question a score in the range 0-10 (left); only three participants in Group 3 got 10 to each question. Distribution of the average time (in seconds) spent in a question by participants per group (right).

selector block configurations on score and time among the three groups and a t-test on the number of clicks on Group 2 and Group 3 to compare the effect of different *CrossWidgets* configurations on the number of clicks. The results are presented in the following.

Score. *Collected on Group 1, 2, and 3.* There was a significant effect of the prototype settings on the score, at the $p < 0.05$ level for the three conditions $F[(2, 25) = 3.82, p = 0.0357]$. To compare samples with different size we have used Fisher's Least Significance Difference (LSD): $LSD_{A,B} = t\sqrt{MS_W(\frac{1}{n_A} + \frac{1}{n_B})}$ where t is the critical value, MS_W is the within mean square obtained from ANOVA and n is the number of scores used to calculate the mean. Post hoc comparison using LSD indicated that the mean score μ_1 for settings of Group 1 was significantly different from the mean score μ_3 for settings of Group 3 ($LSD = 1.24$, $|\mu_1 - \mu_3| = 1.337$), while other differences are not significant. According to that, we can conclude that the Group 3 ($\mu_3 = 8.64$) performed better than Group 1 ($\mu_1 = 7.29$) and we can conclude that the configuration of Group 3 allows a user in getting a higher accuracy in filtering activities than the configuration of Group 1. On the other hand, we cannot make any assumption on differences between Group 1 vs. Group 2 and between Group 2 vs. Group3 because their absolute mean differences are less than LSD. Figure 5 (left) compares scores between Group 1 and Group 3, arranging them by questions. Wrong answers of Group 3 are consistently less than Group 1, except for questions Q3 and Q6. Group 1 clearly performed very bad on question Q5, which appears to be the most difficult one. Figure 5 (right) compares scores between Group 1 and Group 3, arranging them by the task. The number of wrong answers of Group 3 are consistently lower than Group 1 and Group 1 performed very bad in Task 2, which appears to be the most difficult one.

Time. *Collected on Group 1, 2, and 3.* There was **not** a significant effect of the validation prototype settings on Time, at the $p < 0.05$ level for the three conditions $F[(2, 25) = 2.98, p = 0.0687]$ and we cannot make assumptions on difference in time among the groups.

Number of clicks. *Collected on Group 2 and 3.* There was a significant effect of the validation prototype settings on the number of

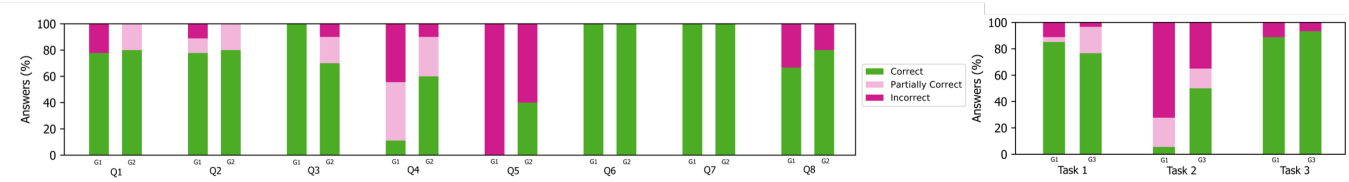

Figure 5: Comparison of the proportion of correct, partially correct, and wrong answers given by participants of Group 1 and Group 3, aggregated per question and per task. Group 2 has been excluded from the comparison because ANOVA pushed us to conclude that there no exist significant effect on scores while comparing it with Group 1 and Group 3. Each question has been given a score in the range 0-10: [0,5) is considered wrong, [5,10) partially correct, and 10 correct.

clicks, at the $p < 0.05$ with $p = 0.020$. According to that, we can conclude that the Group 3 ($\mu = 14.3$) performed better than Group 2 ($\mu = 25.8$) and we can conclude that the configuration of Group 3 allows user to get higher efficiency in filtering activities than the configuration of Group 2. The average number of clicks per task of Group 2 are $T1 = 50.41$, $T2 = 22.67$ and $T3 = 3.37$ while for Group 3 are $T1 = 25.83$, $T2 = 17.15$ and $T3 = 0.97$. The average number of clicks of Group3 is less than Group2 along with the three tasks.

5.5 Discussion

After the analysis of the numerical results and the traces produced by the comparative experiment, the statistical validation that pointed out significant differences, and combining and generalizing the numerical results coming from significant effects (score and number of clicks by questions and by tasks) we are able to present some discussion points and findings.

Single domain selector and filtering activities based on data distribution. The experiment results lead us to conclude that a single selector based on domain performs very bad on tasks that require to set conditions related to the data distribution.

Single selector vs. multiple selectors plus guidance and set based feedback. The experiment confirmed the joint presence for each attribute of a *CrossWidget* with two distinct selectors, one based on the attribute domain and one on the data distribution together with guidance and set based feedback, produces more accurate filtering than using one single selector based on the attribute domain.

Filtering activities based on data distribution. Comparing scores for task shows that Group 3 performed better than Group 1 on filtering tasks requiring to deal with data distribution; however, Task 2 exhibits the highest number of errors and the lowest mean also for Group 3. We can conclude that tasks on multiple attributes are inherently complex independently of the kind of used selectors.

Guidance and efficiency. ANOVA pushed us to conclude that guidance and set based feedback do not significantly increase the accuracy (i.e., the score) of *CrossWidgets* that include domain-based selectors and data distribution selectors. Instead, the significant difference ($p = 0.020$) between the number of clicks of these two configurations pushes us to conclude that guidance and set based feedback lead to a more efficient filtering activity for all the tasks.

Limitations of the proposed approach. Concerning the intrinsic drawbacks of our proposal, we have to consider that our design choices have been compared with traditional sliders. However, an exploration of different alternatives, out of the scope of the paper due to combinatorial explosion of independent variable cardinality - dependent variable cardinality, could produce new insights. As an example, the paper used the box plot as the selector dealing with statistical distribution, but other variations, like, e.g., bargrams, density maps, histograms, or violin plots could be considered, comparing their performances. This aspect might be related to the not statistically significant results raised from the ANOVA validation: we have discussed only significant results, excluding by the analysis very relevant aspects, like answering time. It might be the case that using a relatively new means like the boxplot as a selector contributed to the high variance of response times that made the results statistically less valid. Finally, the paper neglected by design the analysis of different logical connectors both intra and inter the different *CrossWidgets*.

6 CONCLUSION

This paper investigated the activity of filtering, presenting and detailing the concept of *Filtering Overview* and defining the *CrossWidgets* as an attempt to cover with a unique umbrella different proposals and techniques aiming at providing the user with relevant feedback, able to guide her in filtering activities. The main *CrossWidgets* design elements, i.e., attribute domain, data distribution, guidance, and the set-based feedback have been formally characterized, defining their semantics and designing selectors and associated feedback. Several combinations of these means have been hypothesized, identifying a set of viable configurations to be tested, following the different objective of focusing the analysis and the design at *selector level*, explicitly not considering the visualization of the dataset and the associated direct manipulation techniques. These *CrossWidgets* have been implemented in a validation prototype, while a controlled experiment evaluation involving 28 people has been conducted to comparatively assess their advantages and drawbacks. Eventually, a statistical analysis of the collected traces has been conducted. Results seem to confirm the advantages in efficacy and efficiency of the *Filtering Overview* implemented through *CrossWidgets* opening research opportunities in studying additional configurations and providing guidelines for different tasks or datasets properties. According to these considerations, we plan to build on the first positive results we got with this experiment to investigate different means for representing the *CrossWidgets* components, with the goal of comparing different solutions and better elaborate on their performance.

REFERENCES

[1] Christopher Ahlberg and Ben Shneiderman. 1994. Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Boston, Massachusetts, USA) *(CHI '94)*. Association for Computing Machinery, New York, NY, USA, 313–317. https://doi.org/10.1145/191666.191775

[2] Marco Angelini, Graziano Blasilli, Simone Lenti, and Giuseppe Santucci. 2018. STEIN: Speeding up Evaluation Activities With a Seamless Testing Environment INtegrator. In *EuroVis 2018 - Short Papers*, Jimmy Johansson, Filip Sadlo, and Tobias Schreck (Eds.). The Eurographics Association. https://doi.org/10.2312/eurovisshort.20181083

[3] M. Behrisch, F. Korkmaz, L. Shao, and T. Schreck. 2014. Feedback-driven interactive exploration of large multidimensional data supported by visual classifier. In *2014 IEEE Conference on Visual Analytics Science and Technology (VAST)*. 43–52. https://doi.org/10.1109/VAST.2014.7042480

[4] J. Boy, L. Eveillard, F. Detienne, and J. Fekete. 2016. Suggested Interactivity: Seeking Perceived Affordances for Information Visualization. *IEEE Transactions on Visualization and Computer Graphics* 22, 1 (Jan 2016), 639–648. https://doi.org/10.1109/TVCG.2015.2467201

[5] D. Ceneda, T. Gschwandtner, T. May, S. Miksch, H. Schulz, M. Streit, and C. Tominski. 2017. Characterizing Guidance in Visual Analytics. *IEEE Transactions on Visualization and Computer Graphics* 23, 1 (Jan 2017), 111–120. https://doi.org/10.1109/TVCG.2016.2598468

[6] Dheeru Dua and Casey Graff. 2017. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml

[7] Stephen G. Eick. 1994. Data Visualization Sliders. In *Proceedings of the 7th Annual ACM Symposium on User Interface Software and Technology* (Marina del Rey, California, USA) *(UIST '94)*. Association for Computing Machinery, New York, NY, USA, 119–120. https://doi.org/10.1145/192426.192472

[8] S. Gratzl, N. Gehlenborg, A. Lex, H. Pfister, and M. Streit. 2014. Domino: Extracting, Comparing, and Manipulating Subsets Across Multiple Tabular Datasets. *IEEE Transactions on Visualization and Computer Graphics* 20, 12 (Dec 2014), 2023–2032. https://doi.org/10.1109/TVCG.2014.2346260

[9] S. Gratzl, A. Lex, N. Gehlenborg, H. Pfister, and M. Streit. 2013. LineUp: Visual Analysis of Multi-Attribute Rankings. *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (Dec 2013), 2277–2286. https://doi.org/10.1109/TVCG.2013.173

[10] ISO. [n.d.]. ISO 25010 standard. http://iso25000.com/index.php/en/iso-25000-standards/iso-25010/.

[11] Paul Jaccard. 1901. Etude de la distribution florale dans une portion des Alpes et du Jura. 37 (01 1901), 547–579.

[12] Jing Xia, Wei Chen, Yumeng Hou, Wanqi Hu, Xinxin Huang, and D. S. Ebertk. 2016. DimScanner: A relation-based visual exploration approach towards data dimension inspection. In *2016 IEEE Conference on Visual Analytics Science and Technology (VAST)*. 81–90. https://doi.org/10.1109/VAST.2016.7883514

[13] Ian T Jolliffe and Jorge Cadima. 2016. Principal component analysis: a review and recent developments. *Phil. Trans. R. Soc. A* 374, 2065 (2016), 20150202. https://doi.org/10.1098/rsta.2015.0202

[14] D. A. Keim. 2002. Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics* 8, 1 (Jan 2002), 1–8. https://doi.org/10.1109/2945.981847

[15] A. Lex, N. Gehlenborg, H. Strobelt, R. Vuillemot, and H. Pfister. 2014. UpSet: Visualization of Intersecting Sets. *IEEE Transactions on Visualization and Computer Graphics* 20, 12 (Dec 2014), 1983–1992. https://doi.org/10.1109/TVCG.2014.2346248

[16] Qing Li and C. North. 2003. Empirical comparison of dynamic query sliders and brushing histograms. In *IEEE Symposium on Information Visualization 2003 (IEEE Cat. No.03TH8714)*. 147–153. https://doi.org/10.1109/INFVIS.2003.1249020

[17] Peter Rodgers. 2014. A survey of Euler diagrams. *Journal of Visual Languages & Computing* 25, 3 (2014), 134 – 155. https://doi.org/10.1016/j.jvlc.2013.08.006

[18] A. Sarvghad, M. Tory, and N. Mahyar. 2017. Visualizing Dimension Coverage to Support Exploratory Analysis. *IEEE Transactions on Visualization and Computer Graphics* 23, 1 (Jan 2017), 21–30. https://doi.org/10.1109/TVCG.2016.2598466

[19] Hans-Jörg Schulz, Marc Streit, Thorsten May, and Christian Tominski. 2013. Towards a characterization of guidance in visualization. In *Poster at IEEE Conference on Information Visualization (InfoVis)*.

[20] B. Shneiderman. 1994. Dynamic queries for visual information seeking. *IEEE Software* 11, 6 (Nov 1994), 70–77. https://doi.org/10.1109/52.329404

[21] B. Shneiderman. 1996. The eyes have it: a task by data type taxonomy for information visualizations. In *Proceedings 1996 IEEE Symposium on Visual Languages*. 336–343. https://doi.org/10.1109/VL.1996.545307

[22] P. Simonetto, D. Archambault, and C. Scheidegger. 2016. A Simple Approach for Boundary Improvement of Euler Diagrams. *IEEE Transactions on Visualization and Computer Graphics* 22, 1 (Jan 2016), 678–687. https://doi.org/10.1109/TVCG.2015.2467992

[23] Egemen Tanin, Amnon Lotem, Ihab Haddadin, Ben Shneiderman, Catherine Plaisant, and Laura Slaughter. 2000. Facilitating data exploration with query previews: A study of user performance and preference. *Behaviour & Information Technology* 19, 6 (2000), 393–403. https://doi.org/10.1080/014492900750052651

[24] John W Tukey. 1977. Exploratory Data Analysis. In *Addison-Wesley series in Behavioral Science: Quantitative Methods*.

[25] Lisa Tweedie, Bob Spence, David Williams, and Ravinder Bhogal. 1994. The Attribute Explorer. In *Conference Companion on Human Factors in Computing Systems* (Boston, Massachusetts, USA) *(CHI '94)*. ACM, New York, NY, USA, 435–436. https://doi.org/10.1145/259963.260433

[26] C. Weaver. 2010. Cross-Filtered Views for Multidimensional Visual Analysis. *IEEE Transactions on Visualization and Computer Graphics* 16, 2 (March 2010), 192–204. https://doi.org/10.1109/TVCG.2009.94

[27] W. Willett, J. Heer, and M. Agrawala. 2007. Scented Widgets: Improving Navigation Cues with Embedded Visualizations. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (Nov 2007), 1129–1136. https://doi.org/10.1109/TVCG.2007.70589

[28] Kent Wittenburg, Tom Lanning, Michael Heinrichs, and Michael Stanton. 2001. Parallel Bargrams for Consumer-Based Information Exploration and Choice. In *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology* (Orlando, Florida) *(UIST '01)*. Association for Computing Machinery, New York, NY, USA, 51–60. https://doi.org/10.1145/502348.502357

[29] J. S. Yi, Y. a. Kang, and J. Stasko. 2007. Toward a Deeper Understanding of the Role of Interaction in Information Visualization. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (Nov 2007), 1224–1231. https://doi.org/10.1109/TVCG.2007.70515