Setting the Bar for Set-Valued Attributes

Kent Wittenburg

TR2010-030   June 2010

Abstract

In this paper, we present an interactive visualization method for set-valued attributes that maintains the advantages of item-oriented views and interactions found in parallel multivariate visualizations such as bargrams (equal-height histograms). The challenge is to accommodate rendering of an item when it appears multiple times in set-valued attribute views while at the same time preserving value-and item-based selection, brushing, and filtering. Such techniques can help users derive particular types of insights into data based on distributions and correlations of attribute values.

AVI 2010

Setting the Bar for Set-Valued Attributes

Kent Wittenburg
Mitsubishi Electric Research Laboratories, Inc.
201 Broadway
Cambridge, MA 02139
wittenburg@merl.com
1-617-621-7500

ABSTRACT

In this paper, we present an interactive visualization method for set-valued attributes that maintains the advantages of item-oriented views and interactions found in parallel multivariate visualizations such as bargrams (equal-height histograms). The challenge is to accommodate renderings of an item when it appears multiple times in set-valued attribute views while at the same time preserving value- and item-based selection, brushing, and filtering. Such techniques can help users derive particular types of insights into data based on distributions and correlations of attribute values.

Categories and Subject Descriptors

H.5.2 [Information Interfaces and Presentation]: User Interfaces - Graphical User Interfaces; I.3.6 [Computer Graphics]: Methodolgy and Techniques - Interaction Techniques; J2 [Computer Applications]: Physical Sciences and Engineering

General Terms

Measurement, Design, Human Factors

Keywords

Set-Valued Attributes, Information Visualization, Multivariate Visualization.

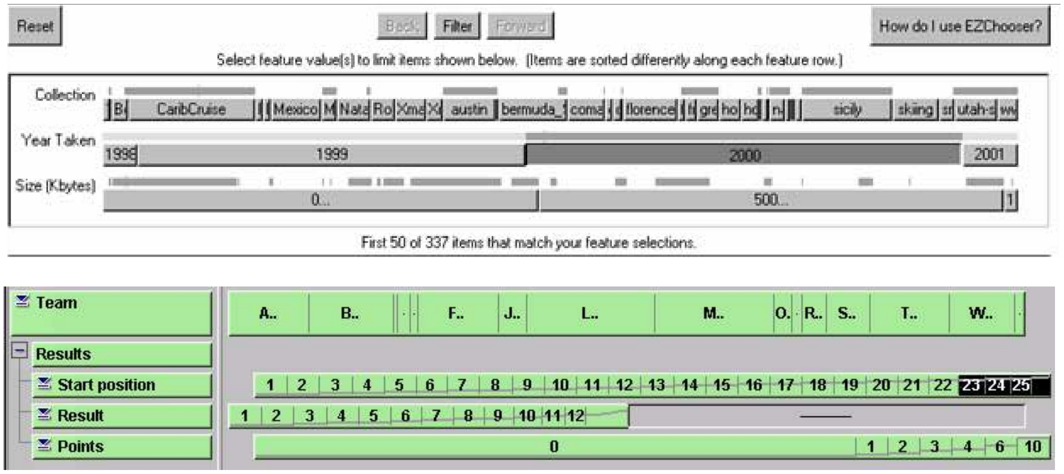
1. INTRODUCTION

A number of visualization methods based on rendering and interaction with attributes in parallel have been proposed for supporting tasks in cases where problems may be initially ill-defined and where human discovery of opportunities and trade-offs is as important as specifying an action [2]. Making an important buying or investing decision, choosing a school to send one's child to or a place to live, specifying a mechanical design with many design parameters, or formulating a company's intellectual property or marketing strategy are but a few examples of such problems.

As long as the attributes in such problem domains can be represented as numeric, ordinal, categorical, or Boolean types, there are a variety of suitable interactive visualization techniques available for presenting attribute information. Examples include dynamic query widgets [1], parallel coordinates [6][3], parallel histograms [10], parallel bargrams (equal height histograms) [8][11][13], paintable sliders [4], and pixel-oriented bar charts [7]. However, in many cases attributes are most naturally represented as set-valued. For instance, the attribute "author" would naturally take as values one or more individuals or text strings. The attribute "keywords" would take one or more keyword values. A product such as a vehicle infotainment system would naturally be represented as one or more members from a set such as {Bluetooth, iPod/MP3 compatible, Navigation, Traffic-info, Premium sound system, Satellite Radio, Audio/Video input, Built-in hard drive, HD radio, DVD player}. None of the visualization methods mentioned above are suitable for such set-valued attributes.

A workaround is to represent each possible value of a set-valued attribute as its own Boolean-type attribute. This technique has the problem that it may miss opportunities to reveal patterns within the distribution of the set-valued attribute considered as a whole. For example, it is often of value to know at a glance how many items may contain all (or most) of the entire set of values. Techniques that create more visually dense widgets for set-valued attributes also may have the virtue of more efficient use of visual real estate. Freiler and Matkovic have recently proposed a method called Set'o'Grams that is visually dense and provides for certain types of insights into the data [5]. They also integrate through brushing other well known visualization methods such as parallel coordinates, scatterplots, and so on, that would be suitable for non-set-valued attribute types. However, the solution proposed by Freiler and Matkovic sacrifices item-oriented views and interactions, which we believe have the significant advantages of simplicity and ease-of-use. By item-oriented views, we mean that individual items are represented graphically on the screen and provide insight as to their distribution wrt attribute information as well as afford direct manipulation for interaction.

In this paper, our goal is to present an item-oriented visual method for set-valued attributes that preserves the ease of use of value- and item-based selection, brushing, and filtering. We will begin by summarizing related work that embodies these advantanges and



Select feature value(s) to limit items shown below. (Items are sorted differently along each feature row.)

First 50 of 337 items that match your feature selections.

Figure 1: Examples of using attribute-ordered items to view distributions by value through equal-height histograms. At the top is EZChooser. At the bottom is InfoZoom. (The latter image is used with permission of Michael Spenke.)

then describe our new technique. Future work includes empirical testing for our proposal as well as dealing with issues of scale.

2. CLOSELY RELATED WORK

A number of methods have appeared in the literature of multivariate visualization in which the entire set of items being viewed can be ordered fully or partially according to each attribute. In the most obvious cases, numeric or ordinal type attributes impose an ordering. Categorical attributes values can be ordered alphabetically and so on. Then value distributions can be shown through bargrams, aka equal-height histograms. Figure 1 shows the technique embodied in some previously published work. Shown at the top is EZChooser [13], which is discussed further in [9]; at the bottom is InfoZoom [11], introduced originally as FOCUS [12], and now a successful commercial product. In contrast to normal tabular views, each row orders the item set differently. The main virtue of this type of parallel attribute visualization is simplicity and generality of both visualization and interaction. It is easy to see what the set distribution is globally over some attribute space. For instance, at the top of Figure 1 it is easy to see that most of the digital photos in this collection were taken in the years 1999 and 2000; in the bottom part, one sees that most of the Formula One race car drivers represented here ended up with 0 points. These rendering methods can be used for most any abstract attribute (except set-valued ones). Interactions that are naturally afforded are filtering the set by selecting a value range through mouse button presses, or, in the case of EZChooser, seeing correlations across values through button presses before filtering [9].

It would be desirable to augment systems such as these with set-valued attributes, but it is not immediately obvious how to do so. If one simply replaced the attributes in a row with the value labels in a set-valued attribute, items would be duplicated in the ordering since they can each have multiple attribute values. The basic conventions would be broken. One could also imagine representing the value row for a set-valued attribute as the power set of values. However, this would explode the cardinality of even moderately sized value sets and make it difficult to render intelligibly.

3. SETTING THE BAR

Our proposal involves displaying the attribute values of set-valued attributes orthogonally, i.e., vertically rather than horizontally. The item set is then ordered consistently across all of its value set members much as in a standard table. Each attribute value has two horizontal areas in its own row. As shown in Figure 2, the top display areas represents the item vector and its selection status. The bottom area contains a bar, possibly with gaps, that represents the distribution of that value across the item set in general, a property functionally equivalent to bargrams for all other attribute types.

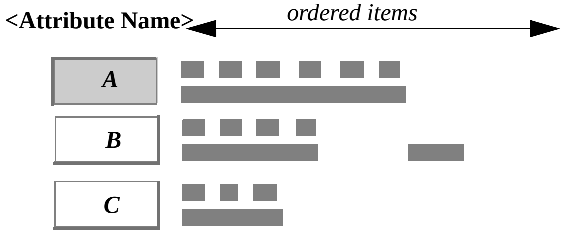
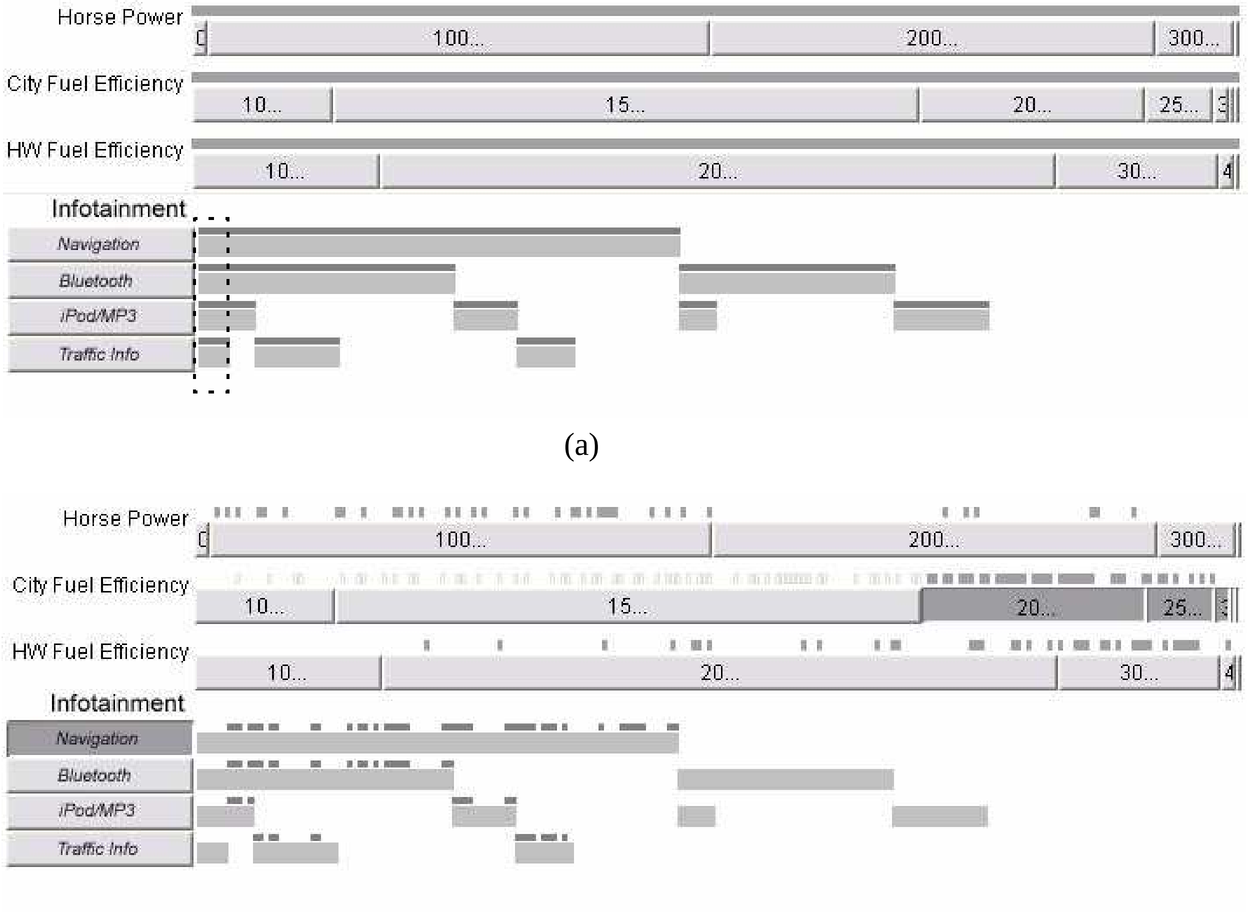


Figure 2: Setting the attribute bar for set-valued attributes. The item vectors are ordered consistently across all set values (A, B, and C). The top row for each value represents its selection status. The bottom row represents the number of positive values in the item set.

For set-valued attributes, we assume the following interaction convention. If a set value is selected, then the item set is restricted to those that include that value in its attribute subset. If more than one set-valued attribute value is chosen, the item set must include the subset of all of the selected values. The implications from Figure 2 are that value A has been selected, thus ruling out some items that have attribute B but not A..

Our proposal for the initial ordering of the elements in a set-valued attribute widget is as follows. First, attribute values are ordered top-to-bottom by frequency across the item set. Second, the item set is ordered left-to-right via the aforementioned attribute value ordering. For example, in a hypothetical dataset of automobiles, let us presume that Navigation was the most frequently occuring value of the attribute Infotainment systems. Bluetooth was next, followed by iPod/MP3 and Traffic Info (see Figure 3). These set values would appear top to bottom in such a sequence. For ordering the items left to right, they are ordered first by whether they have Navigation or not. Then secondarily by whether they have Bluetooth or not, and so on.



(a)

(b)

Figure 3: An example from a fictional car database. In (a), no selections have been made. The view reveals that there is a relatively small number of items having all four Infotainment attributes (dashed rectangle). In (b) a value range for city fuel efficiency has been selected as well as one of the Infotainment attributes (Navigation). Note that these restrictions rule out any Infotainment systems that include both Traffic Info and iPod/MP3. The user may want to explore further.

This inital ordering both vertically and horizontally for set-valued widgets can reveal immediate insights into the distribution of values. In particular, all bars that appear beneath the area of a given bar indicate the distribution of other values with respect to that first value. The top-most value has special status since its bar is contiguous; it is easy to see through vertical alignment roughly how many items that have it as an attribute value also have the others. Because of this preferred status of the top-most attribute set value, one of the interactions we propose is to allow users to drag and drop attribute values vertically to rearrange them. Placing relatively more important attribute values higher in the vertical ordering within a set-valued attribute widget will be visually most informative.

4. EXAMPLE

We now turn to a concrete example to illustrate our proposal. In Figure 3, a hypothetical set of automobiles is shown with the attributes Horse Power, City Fuel Efficiency, Highway Fuel Efficiency, and Infotainment (system). Infotainment is a set-valued attribute with possible values Navigation, Bluetooth, Ipod/MP3, and Traffic Info.

In Figure 3(a), the entire set of automobiles is shown prior to any selection or filtering. From this view one can see the relative distribution over numeric value ranges for the first three attributes in the usual way through bargrams. For the set-valued attribute Infotainment, we can also get a rough idea of the distribution for each of the values by mentally summing up the length of the bars. (It would also be easy enough to show the total figure for the items in each row, as suggested by one reviewer.) We can see that there are relatively few automobiles that contain all four of the Infotainment values as indicated by the dashed rectangle.

Let us presume that our user is interested in cars with relatively good fuel efficiency in the city and he or she is definitely interested in cars with navigation systems. In Figure 3(b), two value restrictions have been made through pressing the relevant “buttons.” Elements in the item vectors change their status. Only those in dark gray remain in the restricted set. From the fact that no items appear in the vertically overlapping areas of the iPod/MP3 and Traffic Info bars, one can deduce that these restrictions rule out cars that have both of these Infotainment values. Perhaps the user wants to relax the fuel efficiency restriction to see what happens? If, upon reflection, Traffic Info and iPod/MP3 are both important, maybe the user should order these attributes first to see how that set of cars fares wrt the other attributes. Perhaps he or she will discover, surprisingly, that there is an automobile that has these highest priority attributes but that also has other unanticipated benefits. On the other hand, the user may be forced to choose one or the other.

This quick and easy "what if" exploration is afforded by our proposal. One of the most fundamental aspects of the discovery task that we defined at the beginning of this paper is to be able to answer such questions as "What will I have to give up to get X?" Through the process of interaction, the user my discover there are dimensions to the problem that they haven't considered. Quick and easy visual explorations of trade-offs before having to filter a global set of choices is important to support this sort of task.

5. CONCLUSION

In this short paper we have first motivated the need to extend a class of multi-attribute visualization techniques to handle set-valued attributes. We have shown some of the benefits that may be afforded by treating the set-valued attribute as a new type of visual widget rather than as a separate collection of Boolean attributes. The main advantage of the method we proposed is that distributions across the set of attribute values can be revealed globally in visual patterns that appear to be easy to understand. Interactions allow users to explore the data as they form decisions related to prioritizing of given set-valued features given the available choices. However, one of the drawbacks of our proposal is that bargrams are split. One of the areas of future work could be a systematic comparison and empirical testing of the method proposed here against a simpler Boolean-type treatment in which bargrams are all contiguous at the cost of giving up consistent ordering of items. In fact, empirical testing in general is needed to verify whether our design concept in fact holds up under closer scrutiny.

When considered against the alternatives, other than what we have just discussed, the only other current proposal for set-valued attributes we are aware of is the Set'o'Gram technique [5]. We believe the advantage of our method is that it maintains the item-orientation of the views. Individual items have a graphic representation in the visualizations, which we believe is important in supporting a class of tasks where selection and filtering of items is of primary interest. Set'o'Grams show distributions of the dataset with respect to the cardinality of set values, but individual items are not rendered nor are they actionable.

As introduced in [13], one of the interesting features of interactive bargrams is the use of "hollow glyphs" in the item vector row to show additional information as restrictions are being made [9]. Instead of an item being just "in" or "out" of the restricted set, it is also possible for an item to be conditionally "in" if the value bar underneath is chosen. This feature can naturally be used with set-valued attributes also. For lack of space, we did not discuss this topic in the current paper but its use is entirely natural here as well.

There are many more issues to be explored in applying the methods described here for databases of even moderately large scale. First is scaling with respect to item set size. This is a common problem for all bargram (equal-height histogram) multivariate visualization methods, and we believe it can be addressed with various forms of data and visual aggregation, many of which are present today in InfoZoom. A separate question is how to scale with respect to attribute set value size. We have encountered databases of interest in which a given set-valued attribute has hundreds of possible values. Methods supporting scaling in the vertical dimension are needed here. Collapsing and expanding over hierarchical attribute trees is, again, already incorporated into InfoZoom, as one can see in Figure 1. However, we believe that there is also a need for other forms of aggregation and visual compression that we leave to future work.

6. ACKNOWLEDGEMENTS

Thanks to Georgiy Pekhteryev for implementation of a prototype of the work presented here and to Robert Spence and anonymous reviewers for valuable feedback and suggestions.

7. REFERENCES

[1] Ahlberg, C., Williamson, C., and Shneiderman, B. 1992. Dynamic queries for information exploration: An implementation and evaluation. In Proc. ACM CHI'92: Human Factors in Computing Systems (1992), 619-626.

[2] Apperley, M., Spence, R., and Wittenburg, K. 2001. Selecting One from Many: The Development of a Scalable Visualization Tool. In Proceedings of IEEE Symposium on Human-Centric Computing Languages and Environments (HCC '01), Stresa, Italy, Sept 2001, 366-372.

[3] Bendix, F., Kosara, R., Hauser, H. 2005. Parallel Sets: Visual Analysis of Categorical Data. Proceedings of the 2005 IEEE Symposium on Information Visualization (InfoVis), 133–140.

[4] Eick, S. 1994. Data Visualization Sliders. In Proceedings of ACM Symposium on User Interface Systems and Technology (UIST '94). 119-120.

[5] Freiler, W. and Matkovic, K. 2008. Interactive visual analysis of Set-Typed Data. In Proceedings of VisWeek '08 (Columbus, Ohio, October 19-24, 2008), IEEE Transactions on Visualization and Computer Graphics 14:6, 1340-1347.

[6] Inselberg, A. 1985. The Plane with Parallel Coordinates.The Visual Computer 1.69-91.

[7] Keim, D.A., Hao, M.C., Dayal, U., and Hsu, M. 2002. Pixel bar charts: a visualization technique for very large multi-attribute data sets. Information Visualization 1:1 (March 2002), 20-34.

[8] Lanning, T., Wittenburg, K., Heinrichs, M., Fyock, C. and Li, G. 2000. Multidimensional Information Visualization through Sliding rods, ACM, Proceedings of Advanced Visual Interfaces (AVI'2000), pp.173-180.

[9] Spence, R. 2007. Information Visualization: Design for Interaction, second edition. Pearson/Prentice-Hall.

[10] Spence, R., and Tweedie, L.1998. "The Attribute Explorer: information synthesis via exploration", *Interacting with Computers*, Vol. 11, pp. 137-146, 1998.

[11] Spenke, M., Beilken, C. 2000. InfoZoom - Analysing Formula One racing results with an interactive data mining and visualisation tool. Second International Conference on Data Mining 2000, 5-7 July 2000, Cambridge University, United Kingdom.

[12] Spenke, M., Beilken, C., Berlage, T. 1996. FOCUS: The interactive table for product comparison and selection. In Proceedings of UIST '96 (Seattle WA, USA, November 1996), ACM Press, 41-50.

[13] Wittenburg, K., Lanning, T., Heinrichs, M., and Stanton, M. 2001. Parallel Bargrams for Consumer-based Information Exploration and Choice. In Proceedings of ACM Symposium on User Interface Systems and Technology (UIST '01) Orlando, Florida, November 2001. 51-60.