Towards a new class of visual analysis systems for space applications

Valery Lavrentiev[1,2]

[1]Technische Universität Berlin
Marchstr. 12, 10587 Berlin, Germany
Mail: v.lavrentiev@campus.tu-berlin.de

[2]Deutsches GeoForschungsZentrum GFZ
Helmholtz-Zentrum, Telegrafenberg, 14473, Potsdam, Germany

Abstract: Data visualization is frequently used as the first step when performing a range of analytical tasks. With the increasing volume and variety of data from space, space science needs new solutions which enable more efficient visual analysis. Existing tools, however, are focused on manual exploration and are limited in their ability to guide users to relevant information. Even though some recommendations exist, they are primarily based on low-level data characteristics and do not consider the user's intent. We plan to extend the paradigm and to take the analytical objective into account. This work will focus on following data tasks as the first step towards the vision: (i) identify; (ii) compare; (iii) summarize. This class of systems can be useful in a variety of space applications, like for example, telemetry, space engineering, and earth observation data analysis.

1. INTRODUCTION

Data analysis is a manual process, where the analyst has to specify the query and the visual encoding in order to understand the structure of the data or find an answer to the target question. If the result does not meet the initial requirements, the process is repeated until desired insights are captured correctly. This is the primary model of interaction for many popular visualization tools [1], [2].

There are many tools which try to address this problem at least partially. On the one hand, there are tools for visualization construction, allowing users to specify the view using high-level grammar [3], but still requiring manual specification. On the other, there are visualization recommendation tools, that automatically suggest interesting data views to the user, but are primarily based on low-level data characteristics and do not take the user’s task into account.

These recommendation tools can be divided into several categories: visual encoding recommenders, data query recommenders and hybrid recommenders. Visual encoding recommendation tools help the user to choose a better visual presentation of the data [4]. Data query recommenders instead suggest interesting data projections with fixed visual encoding. A new class of hybrid recommenders acknowledge the limitations of both and try to implement both visual and data recommendation [5]. Recent studies show that this approach is promising [5].

In contrast to state-of-the-art visual analysis tools, our approach will take the user's intent into account. We suggest to reuse existing practices from software development and recommendation systems and apply them to the user's abstract tasks.

2. RECOMMENDATION ENGINE IMPLEMENTATION

We plan to allow the user to select the required task in the user interface, thus enabling the system to capture their intent directly.

In our research, we focus on strategies to support the following tasks:

2.1 Identify task

An identification task is performed to show interesting properties of a subset of a dataset. For example, a user is looking for a charge input value of a satellite's solar array.

2.2 Compare task

A comparison task focuses on showing differences or similarities between two or more subsets of a dataset. For example, we may compare charge input and voltage across two different time periods.

2.3 Summarize task

A summarizing task focuses on showing information about the whole dataset.

There are many examples of visualizations supporting summarization. The most popular ones can be used in a recommendation engine, e.g. heatmap, splom.

2.4 Data properties

The abstract tasks defined above are referring to data properties or targets, i.e., some aspect of the dataset the user might be interested in. These can be: data distribution, extreme values, trends or specific features, correlation or anything else particularly important to the user. Tasks can be defined as verbs, whereas data properties are nouns.

2.5 Recommendation types

We classify the following recommendation types, which can be used separately or together:

1. **Visual recommendation**

   A visual recommendation suggests possible visual encoding using expressiveness and effectiveness principles [4], [7], [10].

2. **Data recommendation**

A data recommendation ranks interesting subsets of the dataset based on a specific data property.

There are implementations of recommendation systems of this type. For example, the Rank-by-Feature Framework [6] ranks data based on the selected property. The user can choose, for instance, a Pearson correlation metric and the system shows them either the most or least correlated data first.

3. **Related recommendation**

   Similar to Voyager 2 paper [9], a related recommendation suggests alternative visual encoding or adds additional fields in order to enhance understanding of the current focus view. For example, if the user is looking at the solar arrays charge input distribution graph, a related suggestion could be a charge input versus voltage graph.

2.6 Design considerations

The Voyager 2 paper [9] presents design considerations for a faceted chart browser, which we revisit and propose extensions (2) to, focusing on the recommendation part.

1. **Show data variation, not design variation.**

   The system shall focus on showing different data views, instead of visualization encodings.

2. **Allow interactive steering to drive recommendations including an abstract task level.**

   The system shall allow the user to specify their intent directly also on the task level.

3. **Use expressive and effective visual encodings.**

   Recommendations shall apply perceptual design principles [4], [7], [10]

4. **Promote reading of multiple charts in context.**

   Show related charts, so that the effort spent on understanding one chart can aid reading of the next.

5. **Use automation to extend user focus.**

   Base recommendations on the current user context.

6. **Avoid redundant suggestions.**

   Showing many recommendations may distract the user, the system shall limit them by default.

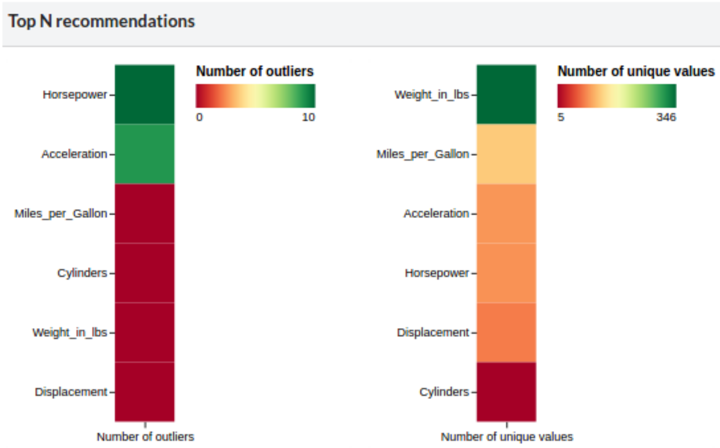

Figure 1: "Top N" recommendation for the summarizing task.

In our implementation, we can combine explicit task and target definition with popular data ranking algorithms in order to provide focused suggestions. When data ranking is complete, we can apply a visual encoding recommendation.

2.7 Usage scenario

Below we describe a possible usage scenario in the future visual analysis recommendation system. In the scope of this example, we assume the following:

1. The system can show the following data properties of a dataset:
    1. Data distribution of a subset.
    2. Number of outliers of a subset.
    3. Pearson correlation coefficient between two subsets.
2. We define for this example the subset as one column (dimension) of the dataset.
3. We use a well-known dataset of automotive statistics [11].

The user starts exploration of the dataset and by default sees the summary screen. The summary screen shows the results of the summarize task – that is, all available data property recommendations. For example, it can show immediately, without any additional interaction, for the properties defined above:

1. “Top N” subsets of the dataset with the most number of outliers
2. “Top N” correlated subsets of the dataset.

Because the system knows that the current task is the summarization task, the above recommendations can be visually encoded to enhance general understanding of the dataset. In figure [1], we use color encoded sorted lists for that purpose. The user might

Figure 2: On the left: "Top N" recommendation for the identification task (sorted left to right). On the right: related view recommendation (bottom).

want to see interesting properties in detail, but is not sure yet what kind of properties or subsets they are interested in. They choose the identify task in the menu.

The system shows similar “top N” lists for the available data properties, but in comparison to the summarization task, each subset is visually encoded to facilitate deeper understanding. In figure [2], the system shows a data distribution histogram for each of the sorted subsets and the scatterplots for the Pearson correlation.

When the user sees interesting data view, they can try to see details about it. The system receives the identify command from the user as well as the subset information. It can recommend them alternative view of the data, either by showing alternative visual encoding or adding additional data dimensions. In figure [2], we enhance detail understanding by related view recommendation.

Finally, the user might want to compare two or more subsets of the data. They do not need to manually choose the comparison function or interesting property. The system can recommend information in the scope of available data properties, but with a focus on showing the difference between the subsets. In figure [3], we use related view recommendation to do that. We can also use data suggestion, for example, by showing “top N” correlations filtered by the given subsets.

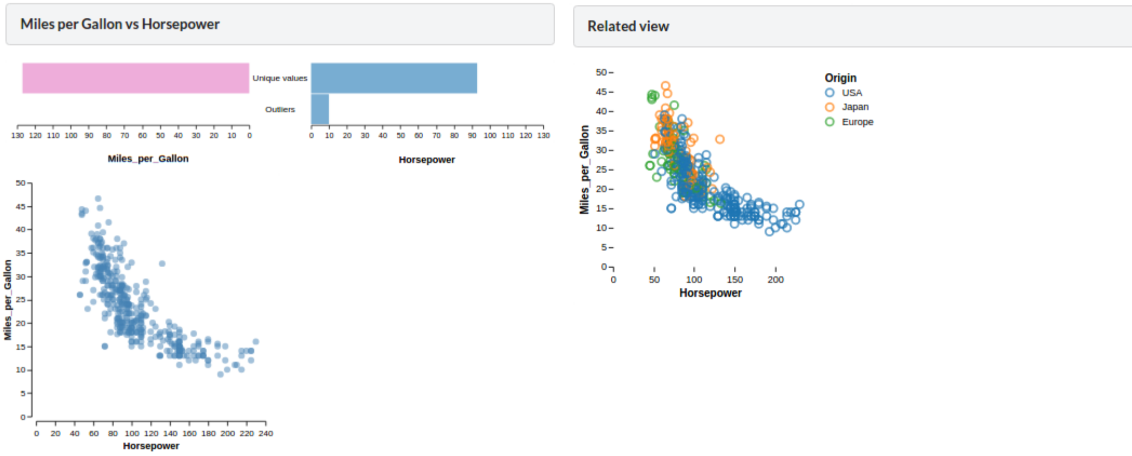

Figure 3: Related view recommendation (right) for the comparison task.

2.8 Findings and future work

We see that the visual encoding and related views recommendation can be applied to every abstract task the user deals with. We can always suggest the user additional relevant data or a suitable visual representation. Data recommendation can be done in every situation where we have a list of subsets and a metric for each of them on a common scale.

Also, we find, that apart from the abstract task specification, the user may want to:

1. Specify a new definition of the data subset for recommendation.

   If we allow the user to specify the subset, e.g, if it is a column, a row or a specific group of columns and rows, then they would be able get automatic suggestions according to this changed definition.

2. Specify a new data property for recommendation.

   The system can recommend the information based on the knowledge of available data properties. In case that there is a way to "tag" or remember a property, that the user wrote themselves, the system can remember it and use it in future suggestions.

We plan to try several solutions to these ideas in our future versions of the system. In addition, we are considering possible ways to gather knowledge about the data, user domain situation and preferences in order to advance the recommendation even further.

We plan to test our prototype system and measure user's performance increase in a few space and general application scenarios.

3. CONCLUSION

Today, many popular applications and websites have recommendation engines powering some of their user's experience. The rise of machine learning and natural language recognition created a path for new applications in the real world, e.g, self-driving cars and speech-based user interfaces. These technologies can be used to gather information about the user's domain knowledge and goals and translate it into machine language in order to assist them in the data analysis task.

Automated assistance of the user's analysis tasks is a complex problem. Splitting it into four cascading levels lets us address different concerns separately: (i) domain situation (ii) data/task abstraction (iii) visual encoding/interaction idiom, and (iv) algorithm. We believe that in order to fully solve the problem, we need to develop solutions to all concerns. In other words, we need to be able to automatically translate user domain knowledge, situation and language into an abstract task definition which is understandable by a machine. Then, depending on the task we should be able to suggest relevant visual encoding/interaction methods and algorithms.

We are implementing the system for the following low-level abstract data tasks first: (i) identify a characteristic in a data set; (ii) compare two or more data sets; (iii) summarize information about all data sets. These actions are building blocks for many user queries during the analysis process. We believe that this research can contribute to development of a "What-Why-How" visualization analysis framework [8]. The goal of the work is to show that the automatic visualization recommendation defined on the user's task level is possible.

4. REFERENCES

[1]: C. Stolte, D. Tang, and P. Hanrahan, Polaris: A System for Query, Analysis, and Visualization of Multidimensional Relational Databases, IEEE Transactions on Visualization and Computer Graphics, 8(1): 52-65, 2012

[2]: H. Wickham, ggplot2: Elegant Graphics for Data Analysis, Springer, 2009

[3]: L. Wilkinson, The Grammar of Graphics, Springer, 2005

[4]: J. Mackinlay, Automating the design of graphical presentations of relational information, ACM Transactions on Graphics, 5(2): 110–141, 1986

[5]: K. Wongsuphasawat, D. Moritz, A. Anand, J. Mackinlay, B. Howe, and J. Heer, Voyager: Exploratory Analysis via Faceted Browsing ofVisualization Recommendations, IEEE Transactions on Visualization And Computer Graphics, 22(1): 649-658, 2016

[6]: J. Seo and B. Shneiderman, A rank-by-feature framework for interactive exploration of multidimensional data, IEEE Transactions on Visualization and Comp. Graphics, 4(2): 96–113, 2005.

[7]: W. Cleveland and R. McGill, Graphical perception: Theory, experimentation, and application to thedevelopment of graphical methods, Journal of the American Statistical Association, 79(387): 531–554, 1984.

[8]: T. Munzner, Visualization Analysis & Design, CRC Press, 2014

[9]: K. Wongsuphasawat, Z. Qu, D. Moritz, R. Chang, F. Ouk, A. Anand, J. Mackinlay, B. Howe, J. Heer, Voyager 2: Augmenting Visual Analysis with Partial View Specifications, 2017

[10]: Jacques Bertin. Semiology of graphics: diagrams, networks, maps, 1983, University of Wisconsin press.

[11]: E.Ramos and D.Donoho. 1983. ASA Data Exposition Dataset. (1983). http://stat-computing.org/dataexpo/1983.html.