Interactive Data Analysis Tool by Augmenting MATLAB with Semantic Objects

Changhyun Lee, Jaegul Choo, Duen Horng (Polo) Chau, and Haesun Park
School of Computer Science and Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332, USA
Email: clee407@gatech.edu, jaegul.choo@cc.gatech.edu, polo@gatech.edu, hpark@cc.gatech.edu

Abstract—The traditional data analysis tools support strong computational capabilities and numerous standard visualization techniques. However, they provide little visual interactions due to the fact that the tools maintain a wide applicability to diverse data domains, and thus any inherent meanings associated with the data domains are hardly allowed. To cover these limitations, we propose to augment Matlab, one of the widely used data analysis tools and computational languages, by imposing the capabilities of handling semantic objects so that diverse essential interaction capabilities could be allowed such as brushing-and-linking, detailson-demand, and dynamic interactive updating on visualization. In our demonstration, we will show our audience how to import semantic data, how visual interactions are occurred, and how these functionalities are convenient using the movie similarity graph data set.

I. INTRODUCTION

The various data analysis tools and computing languages such as Matlab\textsuperscript{1}, R\textsuperscript{2}, and SAS\textsuperscript{3} are available, and these data analysis tools generally provide two main functionalities: computation and visualization. From the perspective of the computational capability, these data analysis tools provide powerful and optimized built-in computational functions as well as their own language that let users to create their own ones. In the context of the visualization, on the other hand, although they provide numerous standard visualization techniques such as a scatter plot, a heat-map, a line graph, and a bar graph, it has limitations to support various visual interactions such as brushing-and-linking that highlights interconnected data items between different views, details-on-demand that enables users to apprehend semantic information of the data items, and interactive updating on visualization that dynamically updates the data. In the case of Matlab, although it provides simple brushing-and-linking interactions, only one-to-one connections are allowed, and connections between numeric matrices are not informative.

One of the main reason for such limitations is that these data analysis tools maintain a wide applicability to diverse data domains, and thus any inherent meanings associated with the specific data domains are not easily allowed. That is, even though the data comes from particular real-world applications such as an image, a document, a gene expression, etc., data analysis tools treat the data in the same way once they are represented as numerical vectors or matrices. As a result, it becomes nontrivial for the data analysis tools to natively support the above-mentioned interactions, which would make sense only when the meanings of data are present in visualizations. To cover these limitations, we propose to augment Matlab, one of the widely used data analysis tools and computational languages, by imposing the capabilities of handling semantic objects so that diverse essential interaction capabilities could be allowed (see Fig 1). We summarize our main contributions as follows:

- Imposing the capabilities of handling semantic objects combined with visualizations so that essential interactions such as brushing-and-linking, details-on-demand, and dynamic interactive updating on visualization can be supported.
- Adding a new set of functionalities and new data type to the widely used data analysis tool, Matlab. Thus, we can adopt highly optimized vector and matrix operations that are already implemented in Matlab.

The proposed capabilities can be applied to other data analysis tools such as R. In this sense, our work has significant potential to be applied immediately in countless analysis problems and domains.

II. THE CONCEPT OF THE OBJECTS

Before introducing the system, we describe the definition of the several objects that we essentially use in the paper.

\footnotesize{http://www.mathworks.com/}
\footnotesize{http://www.r-project.org/}
\footnotesize{http://www.sas.com}
Semantic object The semantic object is any kind of meaningful information. For example, document/image files can be a semantic object, and a comma-separated value (CSV) file can be a set of semantic objects, and even an user-defined description or a title can be a semantic object.

Visual object The visual object is an atomic component of a particular visualization. For example, a cell in a heat-map visualization, a line in a parallel coordinate visualization, and a point in a scatter plot can be a visual object.

Variable object The variable object is general numerical vectors or matrices that are managed by Matlab. The variable object is generally used to visualize a data set.

Users can control the semantic object and the visual object using their given ID, and the variable object is managed by a variable name in Matlab.

III. System Overview

The main purpose of the tool that we propose is assisting visual analytic interactions on Matlab. To achieve this goal, we impose semantic objects that enable communication between any objects. The following subsections describe structure and user interface of the system, the principle of binding between objects that is the core contribution in this paper, and the implementation detail.

A. Structure and User Interface

Fig 2 is the main overview of our system, which has three main panels labeled as (1), (2), (3) and an additional detailed semantic view labeled as (4).

The component (1) in Fig 2 lists the set of semantic objects. The various types of semantic objects such as documents, images, comma-separated value (CSV) file, etc., can be imported. To import the list of semantic objects, the command requires a type of the semantic objects, a title of the list, names of the objects, and a folder path that contains a set of files (.import(type, title, name, url)). Then, the semantic objects are listed as a tabular form and semantic object ID is automatically allocated starting from positive integer 1. Users can import multiple sets of semantic objects, and various interactions such as brushing-and-linking or details-on-demand can be done by clicking a row in a table.

Fig 2(2) visualizes a set of visual objects. This tool provides five essential visualizations: a line graph (.plot), a bar graph (.bar), a scatter plot (.scatter), a heat-map (.imagesc), and parallel coordinates (.pcs). We can create diverse visualizations using command. To bring up the visualizations, the input matrix is basically required and users can adjust visual properties such as colors or sizes by adding name-value pairs properties.
Fig. 3. The semantic objects can be connected to the visual objects. The visual objects can be linked with the indices of the matrix via semantic objects (see visual objects (5,6) and (3,4) in view 2). Users can import multiple lists of the semantic objects and multiple matrices, and multiple connections between objects are generally accepted.

This system supports binding operations that make connections between different objects. There are two types of binding operations (see Fig 3): v-s bind and s-m bind.

The v-s bind binds between visual objects and semantic objects. To bind them, command operation (\texttt{.bind}) requires a set of visual object IDs and corresponding semantic object IDs as arguments. In general, a visual object can be linked with multiple semantic objects. For example, we assume that we visualize a heat-map of the term-doc matrix. A visual object in the heat-map, a cell $(i, j)$, can be referred to the $i$-th term or to the $j$-th document. In addition, any group of the visual objects (an isolated group in a Fig 2 as an example) can also have a name or a user-defined description. Beyond imposing semantic information, these connections enable brushing-and-linking interactions. In fact, the multiple visual objects in different visualizations can be highlighted if the visual objects are inter-linked via the same semantic objects. However, sometimes too many highlights distract visualizations. To avoid this, users have an option to restrict brushing-and-linking functionality in Fig 2 (a). Both detailed semantic information and brushing-and-linking interactions enable users to compare the visual aspects of the visualizations and the actual meaning of them simultaneously with little additional efforts.

The s-m bind binds between column or row indices of a variable object and a list of semantic objects. A command operation (\texttt{.mbind}) requires a variable name, an order of the row or column indices of the variable object, semantic object IDs, and dimension (set dimension to 1 to column-wise or to 2 to row-wise) as arguments. The semantic objects also can be connected to multiple matrix variables. For example, a semantic object is connected to an original input matrix and a dimensionality reduced matrix and a re-ordered matrix. This connections are convenient to extract sub-indices of the matrix. Thus, users can dynamically apply feedback to the input matrix without considering any complicated indexing.

The command operation (\texttt{.mindex}) gives an list of the row or column indices corresponding to the list of semantic objects. This command requires a new variable name, the name of the variable object, semantic object IDs, and dimension (set dimension to 1 to column-wise or to 2 to row-wise) as arguments.

**C. Implementation**

This system is written in Java 1.6 and Matlab 7.13 (R2011). It uses Java Matlab Interface (JMI) for communication with Matlab process. In addition, we embed Beanshell library for dynamic execution of the Matlab syntax in command-line interface. JFreeChart is used to display the high-quality line chart.

https://code.google.com/p/matlabcontrol/wiki/JMI
http://www.beanshell.org/
http://www.jfree.org/jfreechart/
IV. DEMONSTRATION

Datasets We use the pairwise movie similarity matrix obtained from a RottenTomatoes\footnote{http://www.rottentomatoes.com} that is a popular movie rating website. We collect the movie similarity data set by crawling almost 200,000 movies. For convenient analysis, we randomly choose 300 movies that have degrees between 30 and 80 so that a sub-graph is not too sparse nor dense.

Demonstration Details Fig 2 shows the results of the various visualizations of the data set. The heat-map is the clustering result that computed by nonnegative matrix factorization (NMF)\footnote{http://www.tableausoftware.com}, and the scatter plot represents the relationships among movie data items coming from the dimensionality reduction algorithm, t-distributed stochastic neighborhood embedding\footnote{http://www.tableausoftware.com}. The parallel coordinates show the 4-dimensional coefficients that come from NMF algorithm.

In Fig 2, we can find an isolated group on the left-bottom side of the scatter plot. The brushing-and-linking capabilities let users know how the other visual objects in different visualizations reflect these isolated group. As selecting them, we can see that the corresponding visual objects in the heat-map (left-top side) and the parallel coordinates are highlighted. The heat-map (left-top side) shows that the selected movies are strongly connected to each other. In fact, the pop-up view in Fig 2\footnote{4} (4) indicates that the main genres of these movies are 'Animation', and 'Kids & Family', which are preferred by children. The parallel coordinates exhibit that the highlighted lines, a pink color, in the fourth cluster have definitely high scores compared with other lines. It means that the movies of the group are highly inter-connected to each other.

To explore further, we run an existing semi-supervised clustering algorithm based on NMF onto the fourth cluster. To generate constraints, we extract corresponding indices of the selected data items and make constraints so that these movies should be clustered together. This semi-supervised clustering algorithm causes new movies to be joined to the constrained group if they are closely related to the movies in that group.

The heat-map visualization (right-bottom side) in Fig 2 is the result of the semi-supervised clustering algorithm. The clustering result seems to have more strong internal connections. The brushing-and-linking serves that two additional movies are newly joined into this group. Interestingly, one of the movie title that newly joined is 'The Muppet Christmas Carol', which does not belong to the 'Animation' genre. However, the movie is admissible because it is made by 'Walt Disney'.

V. RELATED WORK

The visualization analysis is widely applied to diverse domains. Guess\footnote{E. Adar. Guess: a language and interface for graph exploration. In Proceedings of the SIGCHI conference on Human Factors in computing systems, pages 791–800. ACM, 2006.} is a graph exploration tool with their own command-line interface. It provides several commands such as selecting nodes/edges or changing visual properties that give us some inspiration. Tableau\footnote{J. Choo, C. Lee, C. K. Reddy, and H. Park. Utopian: User-driven topic modeling based on interactive nonnegative matrix factorization. IEEE transactions on visualization and computer graphics, 19(12):1992–2001, 2013.} is a database visualization system that is a commercial successor of Polaris\footnote{C. Stolte, D. Tang, and P. Hanrahan. Polaris: A system for query, analysis, and visualization of multidimensional relational databases. Visualization and Computer Graphics, IEEE Transactions on, 8(1):52–65, 2002.}. Utopian is focused on the document visualization and interactions such as topic splitting, topic merging, etc. All of these tools may be performed well in their own domain, but these tools may not be applicable to other domain.

Kehrer et al.\footnote{3} applies some interactions to the general statistical tool, R, in order to solve statistical problems. The main contribution of the tool is automatically computing statistical algorithms during the data selection and dynamically updating results. In the context of the visual interactivities, this paper is similar to ours, but the presented approach is somewhat limited to specific functionalities, selection and updating.

Our work focuses on enhancing visual inter-activities using semantic objects and augments Matlab in order to take Matlab’s powerful computational capabilities.

VI. CONCLUSION

The general data analysis tools have strong computational capabilities, while it is challenging to have the concept of semantic objects that empower various visual interactivities. Therefore, we propose a new data analysis tool that does not only impose visual inter-activities but also adopts Matlab’s powerful computational capabilities.

We believe that our tool can become an useful tool for scientists and data analysts who are working on analyzing diverse data sets in various domain.

ACKNOWLEDGMENT

REFERENCES