

Augmenting MATLAB with Semantic Objects for an Interactive Visual Environment

Changhyun Lee
Georgia Institute of
Technology
Atlanta, GA 30332, USA
clee407@gatech.edu

Jaegul Choo
Georgia Institute of
Technology
Atlanta, GA 30332, USA
jaegul.choo@cc.gatech.edu

Duen Horng (Polo) Chau
Georgia Institute of
Technology
Atlanta, GA 30332, USA
polo@gatech.edu

Haesun Park
Georgia Institute of
Technology
Atlanta, GA 30332, USA
hpark@cc.gatech.edu

ABSTRACT

Analysis tools such as Matlab, R, and SAS support a myriad of built-in computational functions and various standard visualization techniques. However, most of them provide little interaction from visualizations mainly due to the fact that the tools treat the data as just numerical vectors or matrices while ignoring any semantic meaning associated with them. To solve this limitation, we augment Matlab, one of the widely used data analysis tools, with the capability of directly handling the underlying semantic objects and their meanings. Such capabilities allow users to flexibly assign essential interaction capabilities, such as brushing-and-linking and details-on-demand interactions, to visualizations. To demonstrate the capabilities, two usage scenarios in document and graph analysis domains are presented.

Categories and Subject Descriptors

H.4 [Information Systems Applications]: User Interfaces

General Terms

visual analytics, interactive visualization, clustering, dimension reduction

1. INTRODUCTION

The various data analysis tools and computing languages are available such as Matlab¹, R², and SAS³. These tools

¹<http://www.mathworks.com/>

²<http://www.r-project.org/>

³<http://www.sas.com/>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

IDEA '13, August 11th, 2013, Chicago, IL, USA.

Copyright 2013 ACM 978-1-4503-2329-1 ...\$15.00.

are often geared towards handling generic, typically numerical, data types, and thus they are easily applied in a wide variety of domains as long as the data can be represented as numerical values. These data analysis tools are generally composed of two main capabilities: *computation* and *presentation*. By the former, we mean the capabilities of applying various data processing and transformation techniques to input data while by the latter, we mean those of conveying the input data as well as computed output data to users mainly via *interactive visualization*.

From the perspective of computational capabilities, data analysis tools often support a myriad of built-in computational functions and flexible ways to create users' own ones. It comes to interactive visualization capabilities, even though various standard visualization techniques such as a bar graph, a line graph, a scatter plot, and a heatmap, are supported, little interactions with them are usually allowed. That is, even simple but essential interaction techniques such as brushing-and-linking, which highlights two or more interconnected data items between different visualizations and details-on-demand interactions, which let users explore data details, are hardly provided. Although some tools provide brushing-and-linking, the way of connection between objects is limited and not informative.

The reason for such a problem is mainly due to the fact that these data analysis tools do not inherently assume any semantic meanings associated with the data for the sake of maintaining a wide applicability to different domains. That is, even though the data comes from particular real-world applications such as image, document, 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.

For example, in the context of document analysis, suppose scatter plot is given showing an overall relationship between data points where a data point represent a document. Data analysis tools may maintain only their horizontal and vertical coordinates to draw data points in the scatter plot, but they are not aware of the raw document data

type. Nonetheless, users would still want to explore data points in terms of which documents the data points represent. Furthermore, suppose another visualization is given, say, a bar graph showing the number of documents vs. a year when they were published. One crucial interaction that users want would be brushing-and-linking between different visualizations, say, to analyze when particular data points of interest in the scatter plot were published, etc.

As ways to handle these fundamental needs of users, this paper proposes to augment existing data analysis tools by imposing the capabilities of handling semantic objects so that essential interaction capabilities such as brushing-and-linking and details-on-demand, can be naturally supported. More specifically, from a command-line interface in the data analysis tools, we add a new set of functionalities that enable data analysis tools to maintain a new data type of semantic objects in addition to a generic numerical data type. Furthermore, based on these semantic objects that are explicitly maintained, we provide another set of functionalities that can flexibly assign the interaction capabilities to visualizations.

To realize our idea, we extend Matlab, one of the widely used analysis tools and computational languages, and develop a novel visual analytics system supporting newly proposed command sets while maintaining the full capabilities available in Matlab. The proposed set of commands and capabilities can be expanded in principle 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 where these analysis tools are utilized.

To summarize, the main contribution of this paper includes

- Adding a new set of functionalities and new data type to the existing data analysis tool, Matlab.
- Imposing the capabilities of handling semantic objects with visualizations so that essential interactions can be supported.

The rest of this paper is organized as follows. Section 2 provides the overall description of the proposed system, and Section 3 presents the details of the supported commands in the system. Section 5 discusses related work. Section 4 describes the two real-world analysis examples using the proposed system. Finally, Section 6 concludes the paper and discusses about the future work.

2. DATA ANALYSIS TOOLS AUGMENTED WITH SEMANTIC OBJECTS

Starting with Matlab, the proposed visual analytics system incorporates the concept of *semantic objects*. Examples of semantic objects include any type of meaningful data entities such as documents, images, etc. Similar to the workspace in Matlab where all the variables are maintained, our system has a workspace for semantic objects where semantic objects are created, modified, and deleted, as shown in Figure 1 (1). These semantic objects contain the information about their contents in the forms of text documents or image files that can be accessed when users perform details-on-demand interactions.

Additionally, in order to flexibly associate these semantic objects with visualizations, our system also explicitly handles *visual objects* such as points, lines, axes, etc., composing

a particular visualization. The visual object usually determined by numerical values, and they are represented by the graphical view. For example, a set of dots or cells are displayed on the scatter plot or the heatmap, respectively, in Figure 1.

Given such an explicit management of semantic as well as visual objects, our system provides the functionalities to create links between visual and semantic objects. In addition, the ordinary numerical matrix and/or vector objects from Matlab can also be linked with semantic objects in terms of the matrix/vector row or column indices. For example, each column of a term-by-document matrix, which is generated by a document data set, can be referred as a specific document. Therefore, it can be thought of that there are communication channels between visual object and the index of matrix, which makes it easy to apply visual analytic results to the input matrix.

Next, we will describe the details of the system in terms of the structure and functionalities.

2.1 Detailed Structure and Functionalities

Figure 1 is the main view of our system, which has three main panels labeled as ‘1’ through ‘3’ and an additional popup view labeled as ‘4.’ The first component (Figure 1 (1)) shows the list of semantic objects. The various types of semantic objects, e.g., documents, images, other tabular data such as comma-separated value (CSV) files, etc., can be imported through the command-line interface. In general, the user can handle multiple lists of semantic object sets on the system.

The graphical view panel (Figure 1 (2)) plays a central role in the visualization part. We support five essential graphical views: a line graph, a scatter plot, a bar graph, a heatmap, and parallel coordinates. A dot, which is a part of a line, in the line graph or a dot in the scatter plot corresponds to a visual object. Similarly, a bar in the bar graph or a cell in the heatmap is mapped to a visual object.

Figure 1 (3) is a command-line interface for both Matlab and our own command sets. Once the user issues a particular command on the command-line interface, the interpreter module distinguishes whether the input command belongs to original Matlab command set or our own set we newly defined. To distinguish between the two sets, we attach an additional dot in front of our own command. For example, if an additional dot is put in the beginning of the command-line input, *.imagesc*, our system treats it as our own command set. Figure 2 shows a general workflow of the interpreter module. It begins when the user types some command. Then, this tool interprets whether it is a Matlab command or not. If it is Matlab’s, the system opens communication channel with a Matlab processor. The Matlab executes a received command, then the result is delivered to our system. Otherwise, the system tries to match an appropriate command from our own set of commands and then executes it.

Figure 1 (4) is a semantic-object viewer as a popup window for details-on-demand interactions. For example, Figure 1 (4) shows a tabular form of semantic objects, whose original input data is from a CSV file. Depending on the semantic object type, such as documents, image, a tabular form of data, a corresponding semantic-object viewer opens up.

This tool, whose goal is mainly assisting visual analy-

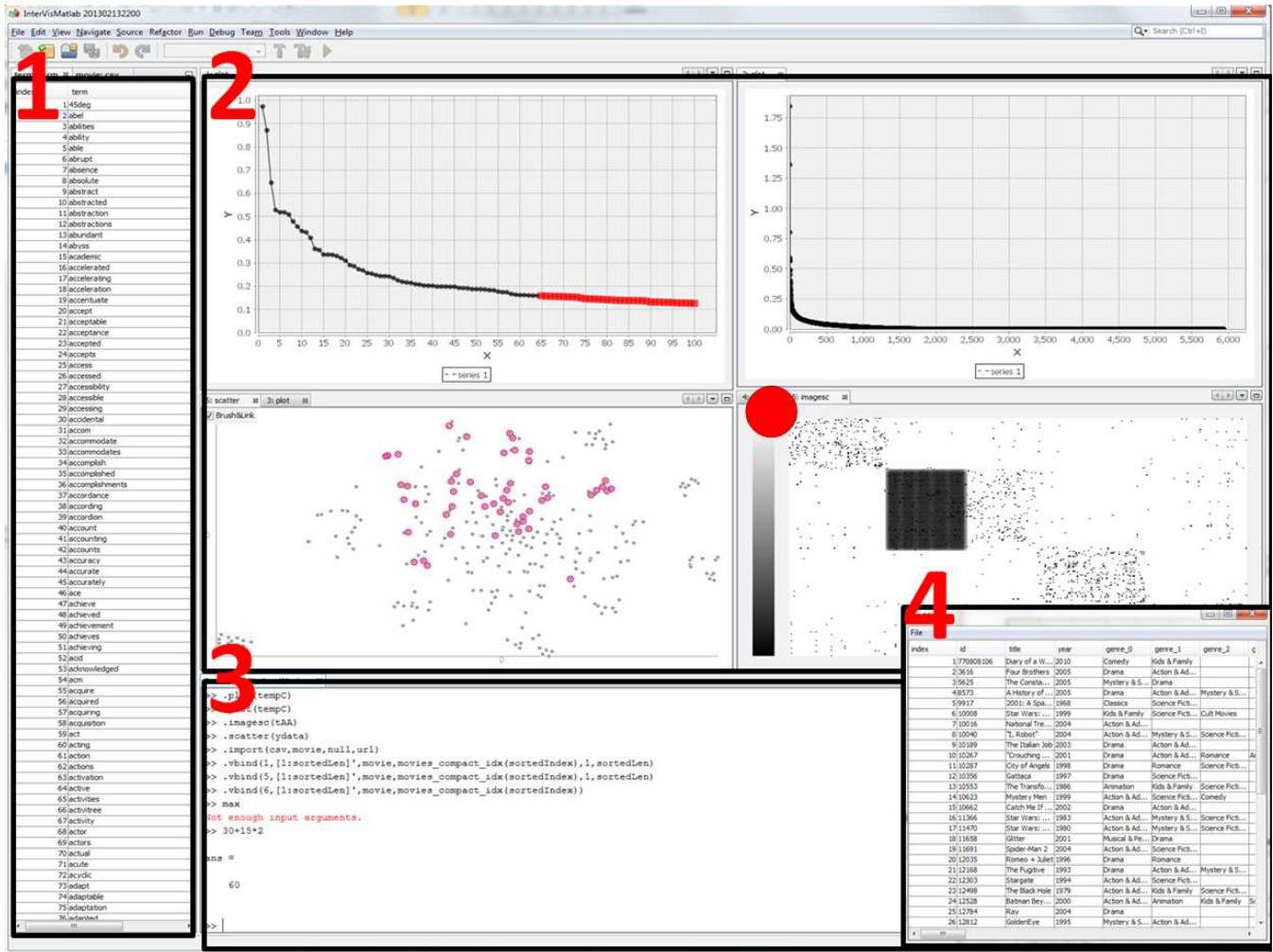


Figure 1: The left-side panel (1) shows the multiple set of semantic objects. The graphical view can be placed in the upper-right side (2). Matlab command or our own command could be executed through the panel placed in the lower side (3). The right-lower side (4) is a semantic-object viewer that shows detailed semantic information. These detailed semantic information can be accessible from visual objects that are linked with semantic objects.

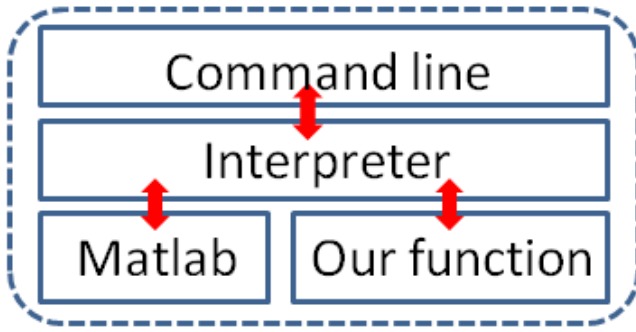


Figure 2: This is a workflow of the command. When the user sends a command, the system interprets it if it is for Matlab or not. If it is for Matlab, it sends command to the Matlab process and then receives computation results. Otherwise, it executes the appropriate function on its own.

sis, provides not only static graphical views but also several interactivities such as multiple linkage options between visual objects and semantic objects, which enable communication between them. It enables the user to look into detailed semantic information along with graphical views as well as to highlight related visual objects by using multiple brushing-and-linking operations. For instance, Figure 1, that is marked as a red circle, represents a heatmap of a term-by-document matrix, and each cell of the heatmap is connected to two semantic objects, a i -th term and a j -th document. In addition, dot representation on the scatter plot and the line graph in Figure 1 represent a dimensional reduction result of some documents and a term distribution of some group of documents, respectively. The user can freely select subset of visual objects by clicking them or dragging a rectangle around them on the plot. The detailed semantic information can be accessible from a semantic-object viewer that pops up by right-clicking the selection. The user can also gain a set of semantic object IDs from the selection. These selected semantic object IDs will be saved in the Matlab as a variable, and the user is allowed to freely use them in the command line. For example, the user can adjust the properties of the visual objects or make additional connections with different semantic objects. The other important functionality, the brushing-and-linking operation, highlight visual objects that linked with selected visual objects via semantic objects. The both detailed information and brushing-and-linking operations enable the user to compare the visual aspects of the graphical view and the meaning of them simultaneously with little additional efforts.

3. COMMANDS

To impose the general capabilities of handling semantic objects, we provide our own set of commands for, say, importing a list of semantic objects and linking between semantic objects and visual objects.

In the command-line interface provided by our system (Figure 1 (3)), the user can seamlessly perform both any Matlab commands and our own commands. The newly defined command sets allow the user to import semantic ob-

jects or to interactively manipulate the properties of the objects.

Next, we introduce our functionalities with comprehensive examples.

3.1 Graphical View

The graphical view is an essential view in visual analytic tools. We provide five well-known graphical views: a scatter plot (`.scatter`), a heatmap (`.imagesc`), a bar graph (`.bar`), a line graph (`.plot`), and parallel coordinates (`.pcs`). Matlab already has these kinds of graphical views, but in order to impose additional interactive capabilities, the proposed system generates our own graphical views independently of Matlab. Each visual object can be connected with multiple semantic objects. In addition, the user can change the properties of the visual object to focus on the subset of important visual objects or for enhanced visual understanding.

To generate views, it requires a $N \times M$ matrix as an input argument. In some cases, N represents the number of visual objects, and M means the dimension of each data item. For example, we need N by M matrix to render N data items that have M -dimensional information on the parallel coordinate plot. However, in the case of heatmap, each cell of a heatmap is represented as a visual object and consequently it draws $N \times M$ visual objects on the view. Every visual object has its own ID which is necessary to bind it with semantic objects. The ID is automatically allocated when creating a view. The heatmap view assigns the visual object ID to each cell using row-major order. Each visual object has various properties and can be adjusted by calling `.setcolor` or `.setsize` functions. The user can also specify properties before creating a graphical view by passing pairs of the Name-Value type argument in the function parameter.

3.2 Bind

We need additional commands for object binding. There are three types of objects: the visual object, the semantic object, and the matrix object. The visual object is presented in the graphical view. The semantic object can be any kind of meaningful information such as a document/image file or a just string or a CSV file. The matrix object is a general matrix/vector variable in Matlab, which represents visual objects on the graphical view.

There are two types of bind operations: `.vbind` for binding the visual and semantic objects, and `.mbind` for binding indices of a matrix object and the semantic objects. `.vbind` requires a list of the visual object IDs and a corresponding list of the semantic object IDs to make correspondences between them. Each visual object can be linked with multiple semantic objects. For instance, we consider that N document data items are visualized on the 2D scatter plot, and the documents are already clustered via some clustering algorithm. Then, each visual object can be referred to as a specific document or a particular cluster it belongs to. Furthermore, if we apply multiple clustering algorithms, we can compare different groups of clusters by connecting multiple cluster memberships. When the visual objects are linked with the semantic objects, other visual objects that are linked with the same semantic object are all inter-linked via same semantic objects. This linkage makes it possible to perform the brushing-and-linking functionality. Next, `.mbind` requires the list of the row/column indices of a matrix and the corresponding list of semantic object IDs. Therefore,

the indices of the matrix and the set of semantic objects are also inter-connected via same semantic objects. Consequently, the user can obtain a subset of indices of the matrix from the visual objects by using a command, `mindex`, that requires a variable name and a set of semantic object IDs. It enables the user to conveniently modify the related matrix of visual objects.

3.3 Implementations

Our tool is currently implemented in Java (version 1.6) and Matlab 7.13 (R2011). To communicate with Matlab, we use Java Matlab Interface (JMI).⁴ In addition, We embed Beanshell library⁵ for enabling dynamic execution of the Matlab syntax. Finally, JFreeChart⁶ is adopted to display the high-quality line chart.

4. ANALYSES

In this section, we show the two different case studies, for a graph data set and a document data set. To highlight our contribution, we represent how it is convenient to analyze the data if the user can easily approach to the meaning of the visual object. In addition, we present the way of data refining and the way of increasing the document clustering quality by making a list of top frequency terms in each cluster.

4.1 Data Sets

First, the graph data set is generated from the pairwise similarity scores obtained from a popular movie rating website.⁷ To be specific, we crawl the information about almost 200,000 movies and make edges between movies if they have a value of similarity. As it is too large to deal with or visualize, we select a subset of 300 randomly chosen movies that have degrees between 30 and 80 so that a sub-graph is not too sparse nor dense.

Second, we use a document data set composed of InfoVis and VAST conference papers from 2001 to 2012.⁸ We build up a bag-of-words representation for each document and build a term-document matrix. The total number of document is 515, and the total number of word is 5,935 after a stopword removal step.

4.2 Usage Scenarios - Movie Data

To gain nontrivial insight about the data, we provide multiple views simultaneously as well as multiple brushing-and-linking functionalities. Figure 3 shows the clustering result of the graph as a heatmap and the dimensionality reduction result of the graph as a scatter plot. The heatmap shows the clustering results that computed by nonnegative matrix factorization (NMF) [6], and the scatter plot shows the relationship among movies that comes from the dimension reduction algorithm, t-distributed stochastic neighborhood embedding [9]. To make clear understanding of the relationship shown in the scatter plot, we apply different colors to visual objects based on the cluster membership.

The set of blue visual objects on the scatter plot is distant from others. Based on the semantic objects, the connected

⁴<https://code.google.com/p/matlabcontrol/wiki/JMI>

⁵<http://www.beanshell.org/>

⁶<http://www.jfree.org/jfreechart/>

⁷<http://www.rottentomatoes.com/>

⁸<http://www.cc.gatech.edu/gvu/ii/jigsaw/datafiles.html>

index	id	title	year	genre_0	genre_1	mpaa_rating
1	9371	Cinderella	1950	Animation	Kids & Family	G
2	9376	The Advent...	1949	Animation	Kids & Family	G
3	9383	Aladdin	1992	Animation	Kids & Family	G
4	9395	Lady and th...	1955	Drama	Animation	G
5	9417	101 Dalmati...	1961	Animation	Kids & Family	G
6	9422	The Rescuers	1977	Action & Ad...	Animation	G
7	9457	The Empero...	2000	Animation	Kids & Family	G
8	9491	Tarzan	1999	Action & Ad...	Animation	G
9	9505	The Rescuer...	1990	Animation	Kids & Family	G
10	9533	The Three C...	1944	Animation	Kids & Family	G
11	9544	Pocahontas	1996	Animation	Kids & Family	G
12	9660	Fun & Fancy...	1947	Action & Ad...	Animation	G
13	9714	Mulan	1998	Drama	Action & Ad...	G
14	9754	Hercules	1997	Animation	Kids & Family	G
15	10291	Lilo & Stitch	2003	Animation	Kids & Family	G
16	10781	Treasure Pla...	2002	Action & Ad...	Animation	PG

Figure 4: The detailed movie information of the isolated group in Figure 3.

columns in the heatmap are highlighted when we select the isolated blue group, as shown in the zoomed-in part in Figure 3. The heatmap shows that the selected movies are strongly connected to each other. In fact, the main genres of these movies are ‘Animation’ and ‘Kids & Family,’ which are preferred by children.

The strength of our tool is that we can use any kinds of the Matlab variables and functionalities. We can obtain the selected column indices of the heatmap, which are the same as the matrix indices. With these column indices, we generate constraints that these movies should be clustered together, which was not the case in the current clustering result. To this end, we run a semi-supervised clustering based on NMF algorithm, which is already implemented in Matlab. This algorithm causes new movies to be included in the constrained cluster if they are closely related to the movies in the constrained cluster.

After running semi-supervised NMF algorithm with the above-describe constraint and a number of clusters as five, we obtain the new clustering result as shown in Figure 5. Clusters are shown to be relatively balanced compared with the previous result. In addition, the last cluster seems to have very strong internal connections. When we open detailed movie information of this cluster, two additional movies, marked as red rectangles in Figure 5, are newly joined into this group. Interestingly, the movie marked with a red circle in Figure 5 does not belong to “Animation”, but the movie is shown to be closely related to “Animation” genre because it is filmed by “Walt Disney”.

4.3 Usage Scenarios - document data

In this scenario, our system shows the topics of document clusters and helps refining the topic of each group. We construct a term-by-document matrix and run a simple NMF algorithm with rank 10 to obtain topic clusters. To verify the quality of the cluster result, we may want to see the term-wise distribution of each group and check how clearly the representative terms represent in each group. The higher value the term has, the more closely the term is related with the group.

For simplicity, we only show four selected cluster results out of 10 cluster results: third, fifth, seventh, and ninth

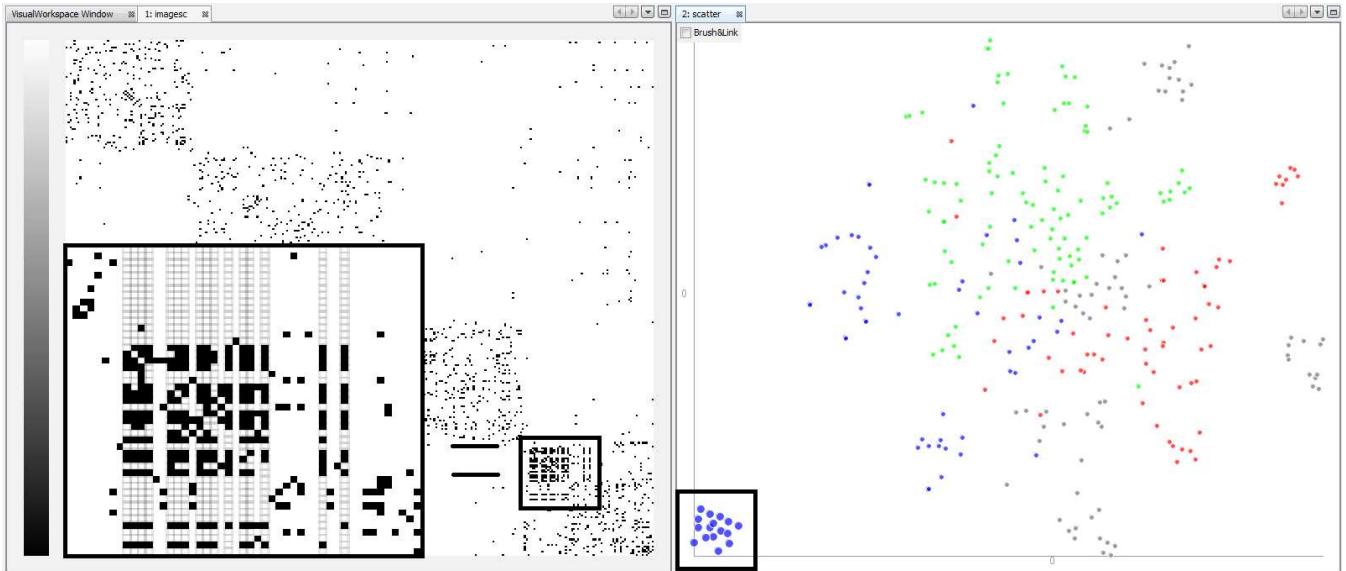


Figure 3: The heatmap represents the clustering result that computed by nonnegative matrix factorization, and the scatter plot represents the dimension reduction result computed by t-distributed stochastic neighborhood embedding.

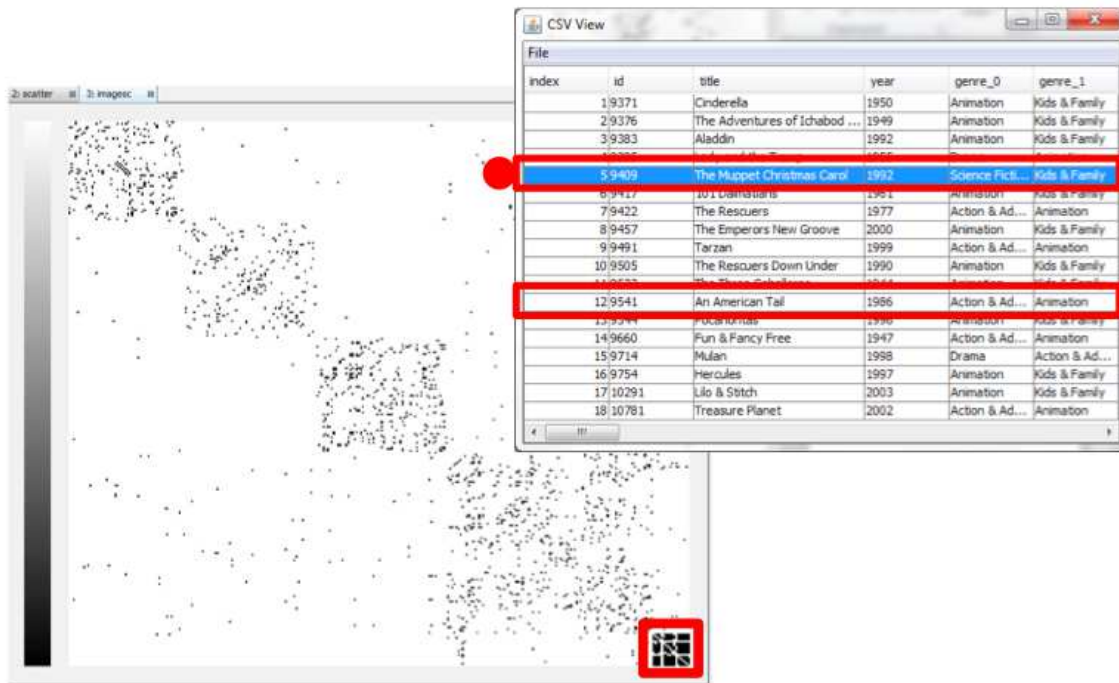


Figure 5: The heatmap represents the semi-supervised clustering result of the movie graph. The semantic-object viewer represents the detailed movie information in the cluster highlighted as a red rectangle. The two movies marked in a red color in the viewer are additionally joined movies to this cluster due to semi-supervision, and the genre of the movie with a red circle does not belong to "Animation" genre while the other movies in the cluster do.

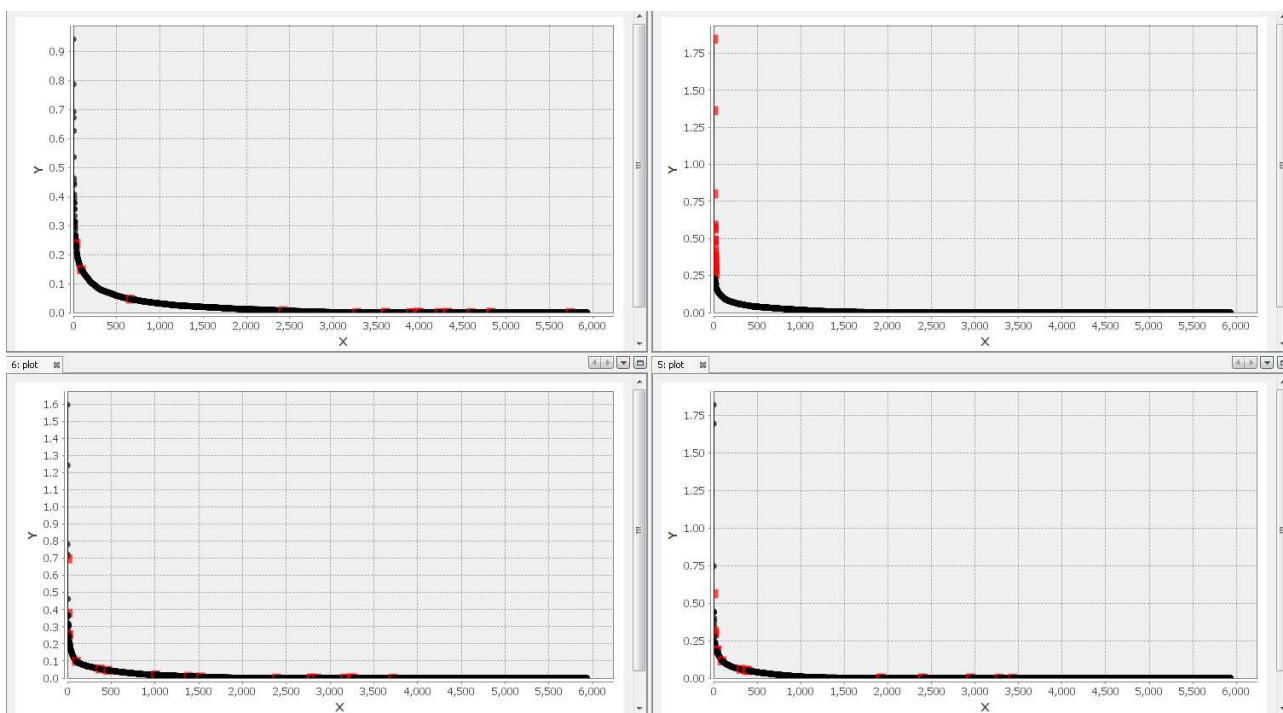


Figure 6: It represents the sorted line graph of term distribution of the group 3 (top-left), 5 (top-right), 7 (bottom-left), and 9 (bottom-right). Top 100 most frequent keywords in the cluster fifth (top-right) are selected and corresponding keywords are highlighted in the other clusters.

Table 1: The three most frequent words in each cluster.

Cluster 3	Cluster 5	Cluster 7	Cluster 9
information	multidimensional	analytics	text
visualizations	dimensions	analytic	documents
context	dimensional	analysis	document

clusters. We draw line graphs of term-wise distributions of topic clusters in an order of decreasing term weights. (Figure 6). This tool binds each value of the line graph with the corresponding term in the dictionary so that the user can easily check which words strongly belong to the specific group. We select top 100 keywords that have the highest values in each group by drawing a rectangle around the line graph. Brushing-and-linking informs the user that corresponding terms have lower coefficients in the other clusters, which explains that each cluster represents distinct topic. Figure 6 shows the one case, the selection of fifth cluster, of brushing-and-linking.

Among the top 100 keywords, we show top three keywords in the Table 1 to understand the topical meaning of each group. The top keywords in each of the seventh and ninth clusters have the same/similar meanings, and thus we merged these words into a single dimension in the term-by-document matrix. In the case of the third cluster, however, the top word "information" does not have much meaning to the user. Therefore, we decided to delete this word from the term-by-document matrix. In this way, we can refine the term-by-document matrix and make the clear topic for each group.

Table 2: The three most frequent words after refining a term-by-document matrix.

Cluster 3	Cluster 5	Cluster 7	Cluster 9
visualizations	dimension	analytic	text
context	parallel	collaborative	document
design	coordinate	visual	corpora

Table 2 represents the top three most frequent words after refining the matrix. In fifth cluster, we can detect more meaningful information such as parallel coordinates. Furthermore, we can see various meaningful words in seventh cluster compared with the previous result.

5. RELATED WORK

In this section, we briefly discuss some of the relevant literature in the context of data analysis tools emphasizing exploratory analysis aspects via interactive visualization. Traditionally, Turkey et al. [8] introduced primitive ways of effective data visualization. Starting from this, many interactive visualization tools have been developed, and some relevant work will be reviewed as follows: NodeTrix [4], Guess [1], iVisClassifier [2], and Tableau ⁹.

NodeTrix [4] visualizes not only overall graph relationships of the data but also detailed adjacency matrix in one view. The main contribution of NodeTrix is that it shows the overall relationships through graph representations as well as detailed relationships through adjacency matrix. It helps the users gain intuitions from both visualizations, but ex-

⁹<http://www.tableausoftware.com/>

tendability is not excellent due to the limited type of graphical viewer and domain. Next, Guess [1] is a graph exploration tool, which allows the user to interact with the graph through its own interpreter, Gython. Guess has a general command-line interface and it gives us some inspiration, but it is only feasible to graph representation. Testbed [3] shows multiple views concurrently to help analyzing the data set. Next, Tableau, which is a commercial successor of Polaris [7], is a database visualization system. However, all of these tools may be appropriate in their own domain, and the main obstacle for these tools is that they require additional learning efforts to use the tools.

The tool that was introduced in Kehrer et al. [5] tries to apply the general statistical tool, R, to the specific functionality, solving statistical problem. The main contribution of the tool is that it automatically computes statistical results during the data selection and dynamically updates the results for the interactive visualization. The basic idea of this paper is similar to ours in that it tries to improve the interactivity of general statistical tool, but the presented approaches are somewhat limited to the generic data types and specific functionalities, selection and updating.

In this paper, we suggest the tool that imports various data types in Matlab environment in order to seamlessly generalize and enhance the interactivity with visualizations.

6. CONCLUSION AND FUTURE WORK

Previously, it was challenging to have the concept of semantic objects originated from application domains in general-purpose data analysis tools. Some domain-specific tools may naturally provide various interaction capabilities applicable to the particular data type. However, it is not always able to be generalized to other domains, and thus the user should consider various tools depending on the domain. Therefore, starting from one of the widely-used tool, Matlab, we have proposed binding capabilities between general semantic objects and visual objects for helping visual analysis. In addition, our system does not compromise any of the Matlab functionalities, and thus the user can still use any kinds of Matlab functionalities and commands.

There are still a few issues to be improved in our system. We use command-line interface, but it usually needs some time to get used to. Thus, we plan to implement an additional window panel that visualize summary of current binding information. In addition, we plan to refine the user interfaces, especially by improving the binding operation into easier GUI interfaces.

7. ACKNOWLEDGMENTS

The work of these authors was supported in part by the National Science Foundation grant CCF-0808863. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

8. REFERENCES

[1] E. Adar. Guess: a language and interface for graph exploration. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '06*, pages 791–800, New York, NY, USA, 2006. ACM.

[2] J. Choo, H. Lee, J. Kihm, and H. Park. ivisclassifier: An interactive visual analytics system for classification

based on supervised dimension reduction. In *Visual Analytics Science and Technology (VAST), 2010 IEEE Symposium on*, pages 27–34. IEEE, 2010.

[3] J. Choo, H. Lee, Z. Liu, J. Stasko, and H. Park. An interactive visual testbed system for dimension reduction and clustering of large-scale high-dimensional data. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*. To appear, 2013.

[4] N. Henry, J.-D. Fekete, and M. J. McGuffin. Nodetrix: a hybrid visualization of social networks. *Visualization and Computer Graphics, IEEE Transactions on*, 13(6):1302–1309, 2007.

[5] J. Kehrer, R. N. Boubela, P. Filzmoser, and H. Piringer. A generic model for the integration of interactive visualization and statistical computing using R. In *Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on*, pages 233–234. IEEE, 2012.

[6] F. Shahnaz, M. W. Berry, V. P. Pauca, and R. J. Plemmons. Document clustering using nonnegative matrix factorization. *Information Processing & Management*, 42(2):373–386, 2006.

[7] 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.

[8] J. W. Tukey. Exploratory data analysis. *Reading, MA*, 231, 1977.

[9] L. van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.