LSAView: A Tool for Visual Exploration of Latent Semantic Modeling

Patricia J. Crossno* Daniel M. Dunlavy† Timothy M. Shead‡

Sandia National Laboratories

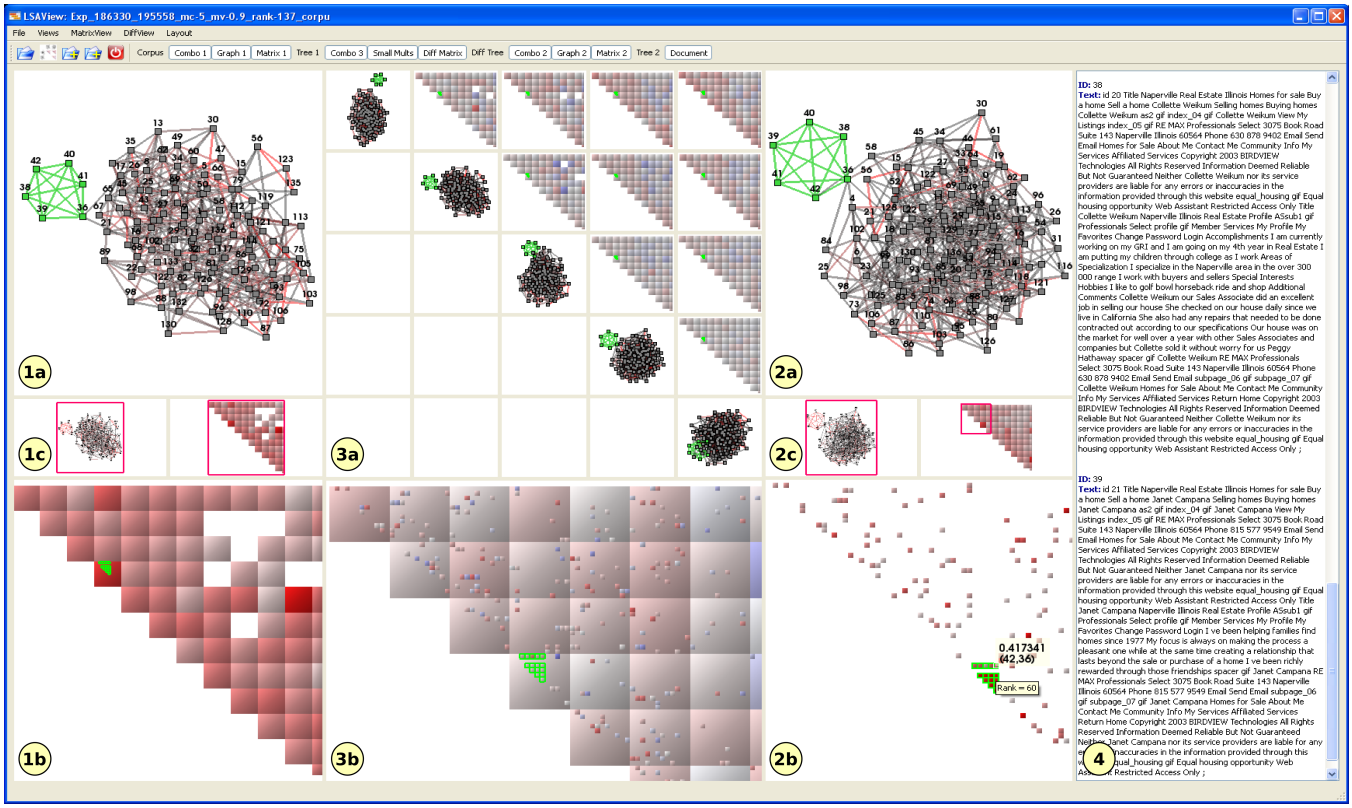

Figure 1: Examples of different views in the LSAView application: (1a) and (2a) GRAPH VIEW, (1b) and (2b) MATRIX VIEW, (1c) and (2c) YOU ARE HERE VIEW (3a) SMALL MULTIPLES VIEW, (3b) DIFFERENCE MATRIX VIEW (4) DOCUMENT VIEW. The CORPUS VIEW and TABLE VIEW are not shown here.

ABSTRACT

Latent Semantic Analysis (LSA) is a commonly-used method for automated processing, modeling, and analysis of unstructured text data. One of the biggest challenges in using LSA is determining the appropriate model parameters to use for different data domains and types of analyses. Although automated methods have been developed to make rank and scaling parameter choices, these approaches often make choices with respect to noise in the data, without an understanding of how those choices impact analysis and problem solving. Further, no tools currently exist to explore the relationships between an LSA model and analysis methods. Our work focuses on how parameter choices impact analysis and problem solving. In this paper, we present LSAView, a system for interactively exploring parameter choices for LSA models. We illustrate the use of LSAView's small multiple views, linked matrix-graph views, and data views to analyze parameter selection and application in the context of graph layout and clustering.

Index Terms: I.3.8 [Computing Methodologies]: Computer Graphics—Applications; I.2.7 [Computing Methodologies]: Natural Language Processing—Text analysis

1 INTRODUCTION

Automated processing, modeling, and analysis of unstructured text (news documents, web content, journal articles, etc.) is a key task in many data analysis and decision making applications. In many cases, documents are modeled as term or feature vectors and latent semantic analysis (LSA) [6, 7, 16] is used to model latent, or hidden, relationships between documents and terms appearing in those documents. LSA supplies conceptual organization and analysis of document collections by modeling high-dimension feature vectors in many fewer dimensions. In this paper, we concentrate on how parameter choices used in LSA impact document relationship modeling in the context of graph layout and clustering methods.

LSA computes a truncated singular value decomposition (SVD)

*e-mail: pjcross@sandia.gov
†e-mail: dmdunla@sandia.gov
‡e-mail: tshead@sandia.gov

of a term-document matrix [3], i.e., the collection of feature vectors associated with the documents in a text collection, or corpus. More specifically, the rank-k LSA model of a term-document matrix, $A \in \mathbb{R}^{m \times n}$, is its rank-$k$ SVD,

$$A_k = U_k \Sigma_k V_k^{\mathsf{T}} \ , \tag{1}$$

where $U_k \in \mathbb{R}^{m \times k}$, $\Sigma_k \in \mathbb{R}^{k \times k}$, $V_k \in \mathbb{R}^{n \times k}$ contain the k leading left singular vectors, singular values, and right singular vectors, respectively. Furthermore, $U_k^{\mathsf{T}} U_k = V_k^{\mathsf{T}} V_k = I_k$, where I_k is the $k \times k$ identity matrix. Often, the rank of the LSA model in (1) is chosen such that $k << \min(m,n)$, leading to a reduction in model noise and computation for many analysis methods.

One particular type of analysis that is widely performed using LSA—and the motivating application for the work presented in this paper—is determining conceptual relationships between two documents, two terms, or a term and a document. Graph data structures and algorithms are often used in this case [15]. For example, document clustering using graph layout methods and LSA modeling can be performed by first computing distances, or similarity scores, between all pairs of documents using the right singular vectors of the rank-r SVD of a term-document matrix. In this work, we use cosine similarities, defined as

$$e_{ij}(k) = \frac{\langle v_k^i \Sigma_k, v_k^j \Sigma_k \rangle}{\|v_k^i \Sigma_k\|_2 \, \|v_k^j \Sigma_k\|_2} \ , \tag{2}$$

between documents i and j, where $\langle \cdot, \cdot \rangle$ is the standard inner product, v_k^i is the ith row of V_k from (1), and $\| \cdot \|_2$ is the L^2-norm, or standard Euclidean distance. The similarities are stored as a similarity matrix, E, whose element (i,j) is defined in (2). To support large corpus analysis, only edge weights above a threshold are used in practice, leading to sparse similarity matrices. This similarity matrix is then used as a weighted adjacency matrix to construct a similarity graph. In this graph, nodes represent documents and edges represent the relationships between documents, weighted by similarity scores. Finally, graph layout methods are used to represent clusterings of the documents, i.e., related nodes are grouped together and unrelated nodes are separated in the resulting graph layout.

A central challenge when using LSA for text analysis is determining appropriate parameters for the SVD, particularly selecting the rank of the SVD and scaling of the singular values for different data and types of analysis. The rank selection problem refers to the determination of an appropriate rank of the truncated SVD for use with a particular task and data set. For the problem of document clustering, a suitable rank is typically determined by analyzing document sets related to the collection to be clustered. Clusterings for these related collections are used to tune the LSA rank parameter for the collection to be clustered [17]. This approach requires annotated document collections whose term-document relationship distributions are highly correlated with those of the document collection to be clustered. Such annotations are laborious to generate and the results may contain errors or subjective clusterings. Thus, a new technique for solving the rank selection problem is needed for the problem of document clustering.

Our approach for solving the rank selection problem in document clustering uses visual comparisons of the impacts on document groupings in the model resulting from parameter changes to LSA. In particular, we look for the formation of document groups with more heavily weighted edges within clusters and lightly weighted, or non-existent, edges between clusters.

Singular value scaling (or rescaling) refers to an exponential scaling of Σ_k by $\alpha/2 \in \mathbb{R}$. The result of singular value scaling can be characterized as a contraction ($0 < \alpha < 2$), expansion ($\alpha > 2$), inversion ($\alpha < 0$) or flattening ($\alpha = 0$) of the singular value spectrum. In the original LSA work in information retrieval, the value of $\alpha = 2$ (i.e., no scaling) was used [7], whereas subsequent research demonstrates the usefulness of contractions and inversions of the singular value spectrum [3]. The choice of scaling often varies depending on application: for example, inverting the singular values tends to highlight novel and anomalous relationships when clustering documents.

For the document clustering problem, the use of singular value scaling changes the similarity scores in (2) to

$$e_{ij}(k,\alpha) = \frac{\langle v_k^i \Sigma_k^{\alpha/2}, v_k^j \Sigma_k^{\alpha/2} \rangle}{\|v_k^i \Sigma_k^{\alpha/2}\|_2 \, \|v_k^j \Sigma_k^{\alpha/2}\|_2} \ . \tag{3}$$

Although originally developed to improve information retrieval systems [1, 27], singular value scaling can be used for any analysis task employing LSA models. However, as with the rank selection problem, no tools exist for visually exploring the relationships between the scaling parameter and analysis methods.

In this paper we present LSAView, a system for interactive exploration of the impact of parameter choices in graph-based informatics analysis systems on the visual presentation and analysis capabilities that data analysts utilize in decision making processes. Specifically, we present the visualization capabilities of LSAView and illustrate how they can be used to understand the relationships between parameters used in LSA and in the application area of graph-based document cluster analysis. LSAView fills a gap for algorithm developers who require better understanding of the impact and sensitivities of parameters in their methods and for data analysts who need to better understand the models used in their analyses. Through visual exploration both developers and analysts can investigate the complex relationships between algorithms, models and analysis.

The major contributions of this work are as follows:

- A framework for visually exploring the relationship between LSA model parameters and graph clustering methods.
- A new MATRIX VIEW to support scalable, zoomable visualization of matrix and matrix difference data.
- Visualization of graph statistics to identify unexpected edges associated with LSA model parameters.
- Case studies illustrating the use of visual algorithm analysis to identify the impact of LSA model parameters on graph layout and clustering methods.

The remainder of this paper is organized as follows. In Section 2, we discuss related work in the areas of LSA and visualization. LSAView is described in Section 3. Section 4 illustrates the use of LSAView in two case studies, and conclusions are presented in Section 5.

2 RELATED WORK

2.1 LSA: Rank Selection and Singular Value Scaling

The method of LSA presented in the previous section has led to many variants that show promise across a broad range of applications: e.g., probabilistic LSA [12], LSA using the semi-discrete [14] and non-negative [2] matrix decompositions, and LSA-like tensor decompositions [13]. Common to all of these methods are the choices of the dimension of the latent feature space (i.e., rank of the matrix or tensor decomposition) and scaling of the latent features (i.e., singular value scaling in the case of the SVD-based LSA method). Therefore, progress made in solving the rank selection and singular value scaling problems described in the previous section will likely lead to analogous improvements in these related methods. In this paper, though, we restrict our attention to the SVD-based LSA method.

A recent survey indicates the challenges associated with rank selection for the problem of LSA and provides many examples of "optimal" rank settings for a wide variety of problems [4]. Statistical

and probabilistic methods for selecting the rank of an SVD in data analysis applications include cross validation [20, 25], Bayesian model selection via Markov chain Monte Carlo methods [11], expectation maximization [22], and Bayesian inference [18]. However, these methods focus on rank determination with respect to noise in the the data and not with respect to how the SVD will be used for analysis. As an example of the potential shortcomings of these methods, we compare LSAView and a cross validation method in the case studies in Section 4. Attempts to determine a useful SVD rank for specific problems exist [15], but tools for general exploration of relationships between the rank of the SVD and its impact on text analysis methods do not.

Similarly, methods for scaling the singular values have been developed for information retrieval applications [1, 27], but no tools for exploring the relationships between spectral properties and analysis methods currently exist.

2.2 Visualization

Eick, et al. developed a multi-view system for evaluating supervised learning algorithms used in solving computational linguistics problems [8]. After training the system, they measured classification accuracy against a set of predefined categories and concepts. Outside text analysis, Groth described a multi-view system to examine the performance of a naive Bayes classifier with respect to various discretization schemes [10]. While these systems provide visualization and analysis of the performance of individual models, neither provides comparative analyses of multiple models.

3 LSAVIEW

LSAView is built using Sandia National Laboratories' open-source Titan Informatics Toolkit [5, 26]. It uses a multiple-coordinated-view approach to explore the impact of parameter choices, as shown in Figure 1. The application's input data consists of a corpus with the text for each document, and one or more document similarity graphs, each produced using different values of the LSA parameters (e.g., the LSA model rank and/or singular value scaling parameters highlighted in this paper).

The LSAView views were designed to present different visualizations of individual analysis models, along with comparative visualizations of collections of models. The GRAPH VIEWS display similarity graphs using typical layout methods, while the MATRIX VIEWS use color-coding to permit visual comparisons of edge weights, statistics calculated over a range of result graphs (see Section 3.2.2), and explicit differences in values or statistics between matrices associated with two graphs. Non-graphical table views enable drill-down to explicit numerical values for the edge weights and statistics for each edge. Text-based views include the CORPUS VIEW and DOCUMENT VIEW to enable examination of source texts for selected documents.

The graphical views are grouped into three panels. In Figure 1, these panels are enabled along with the DOCUMENT VIEW on the far right. The left and right graphical panels provide detailed inspection of two document similarity graphs resulting from different LSA models. The upper GRAPH VIEWS and lower MATRIX VIEWS each represent the same data in a different form. In between, two YOU ARE HERE VIEWS provide context and an alternative navigation method within the corresponding window.

The middle panel contains a SMALL MULTIPLES VIEW at the top and a DIFFERENCE MATRIX VIEW at the bottom. The DIFFERENCE MATRIX VIEW shows color-coded differences between the left and right panels' edge weights or other statistics derived from the adjacency matrices associated with different LSA models. The SMALL MULTIPLES VIEW provides a high level view of graph layouts and differences between adjacency matrices for up to five different LSA models, which need not match the models that are shown in the side panels. To assist in managing this large collection of views, a series of display toggle buttons for both individual views and for entire panels is provided in the toolbar at the top of the application. Additionally, double-clicking within a view will expand it to fill the entire application window, simultaneously turning off all other views. Double-clicking again will restore the previous view configuration. For GRAPH VIEWS or MATRIX VIEWS with an associated YOU ARE HERE VIEW, double-clicking includes the YOU ARE HERE VIEW in the expanded view.

3.1 Graph Views

Each GRAPH VIEW displays a similarity matrix E as a node-link diagram, where nodes represent documents and edges represent the weighted similarities between documents, as defined in (2). The graph provides a high-level view of how document clusters change relative to parameter changes. In the SMALL MULTIPLES VIEW in Figure 1, the five GRAPH VIEWS show the impact of changing rank on the connectivity of a document group, as demonstrated by the highlighted small cluster of documents that is not connected to the corpus using a lower-rank LSA model (top left) but is connected when using a higher-rank LSA model (bottom right). Since the choice of graph layout algorithm can impact the perceived groupings, LSAView users may select from among the many layout algorithms provided by Titan. All of the figures in this paper use `vtkFast2DLayoutStrategy`, a density-grid-based layout algorithm that executes in linear time in the sum of nodes and edges of the graph.

Edges are color-coded using saturation to denote similarity values, with low values in gray and high values in red. Nodes, edges, or both edges and nodes contained in a rectangular region can be selected. Selections are highlighted in green and are linked to all other views. Note that not all views share the same edges, so each view is limited to highlighting the selected edges that are shared across views.

3.2 Matrix Views

3.2.1 Visualization

Each MATRIX VIEW provides a scalable visualization of a similarity matrix E or a matrix of statistics on the similarity values. Conceptually, each value in the matrix (e.g., the similarity value of each edge in the corresponding similarity graph) is rendered as a small rectangle. Since the size of the matrix (the number of documents in a corpus) will exceed the screen resolution for all but the most trivial of corpora, some mechanism is necessary to provide navigation through subsets of the matrix. Naive in-and-out zooming of the matrix was deemed insufficient, since "zooming out" to display the entire matrix leads to sub-pixel rendering of the individual values, leaving little useful context for navigation. To provide a usable visualization with useful summarization when "zoomed out", the MATRIX VIEW provides treemap-like "levels of detail", binning individual matrix values into larger rectangular "bins". The binning process is applied recursively, creating larger-and-larger bins until we are left with a single all-encompassing bin at the root of a tree. With the individual matrix values thus organized, we can render bins at any level in the tree based on the current zoom level, with each bin efficiently summarizing its contents.

For example, the bins in a MATRIX VIEW associated with a single document similarity graph can be used to visualize edge weights or one-sample t statistics on the edge weights. When viewing edge weights, different options are available for summarizing the values within a bin, including minimum, maximum, and average child weight. For the t statistics, the maximum value of each bin's children is used as its summary value.

For DIFFERENCE MATRIX VIEWS, a matrix of values derived from a pair of similarity graphs is first computed, including two-sample t statistics, as well as differences (edge weights and sample

means) and summations (standard errors) of the elements of the two adjacency matrices associated with the two graphs. The resulting matrix of differences is then rendered as in a MATRIX VIEW. In all DIFFERENCE MATRIX VIEWS, the maximum value within a bin is propagated upward.

Bins and values are color-coded to permit visual comparisons between graphs. All of the values to be color-coded, except for the t statistics, range from negative one to positive one. Saturation is used to show increasing absolute value, with zero encoded as white. Large positive values are shown in bright red, large negative values in deep blue, and the lookup table is constructed using linear interpolation. Selected bins and values are outlined in green to stand out against the blue-red palette, and selections are linked to all other views.

For the t statistics (see Section 3.2.2), we are interested only in the magnitudes of the values, so we use saturation ranging from white to bright green to encode increasing absolute value. The lookup table is constructed using a log scale to focus color and attention on the highest values in the matrix. Selected bins and values are outlined in red to stand out against the green palette, and selections are linked to all other views.

The initial rendering of the MATRIX VIEW displays the entire tree at the lowest level that preserves a minimum rectangle size. This facilitates the use of a MATRIX VIEW at all scales, including within the SMALL MULTIPLES VIEW (Section 3.4). The user interface is zoomable so all levels of the tree are accessible. As the user zooms into a deeper level, the rendering of the new level overlaps the old level to provide context. Once the size of the rectangles at the lower level exceeds a threshold, the higher level rectangles are no longer rendered. The DIFFERENCE MATRIX VIEW in Figure 1 demonstrates overlapped rendering of two tree levels, while the MATRIX VIEW to its right demonstrates the appearance of a view after zooming all the way in to the leaf level of the tree (note that at the leaf level, the sparsity pattern of the matrix becomes apparent, since actual values are rendered, instead of bins).

Numeric values for each edge and summary values for each bin are displayed in a "hover balloon" as the mouse moves over a rectangular region. The region associated with the value is highlighted in white to visually confirm the position being referenced. The application will always select the lowest node (smallest encompassing rectangle) on the tree associated with the mouse position.

3.2.2 Matrix Data

We define the sample mean of $e_{ij}(k,\alpha)$ using $n+1$ samples as

$$\bar{e}_{ij}(k,\alpha,n) = \frac{1}{n+1}\sum_{r=k-n/2}^{k+n/2} e_{ij}(r,\alpha)\ , \tag{4}$$

and the corresponding standard error as

$$s_{ij}(k,\alpha,n) = \sqrt{\frac{1}{n}\sum_{r=k-n/2}^{k+n/2}\left(e_{ij}(r,\alpha)-\bar{e}_{ij}(k,\alpha,n)\right)^2}\ , \tag{5}$$

where α is the singular value scaling parameter. Note that the statistics use biased samples centered around edge weights associated with a rank-k LSA model. The purpose of these statistics are to help identify anomalous edge weights given variances in those weights across the most closely related LSA models.

Using the sample mean and standard error definitions above, we define a one-sample t statistic with sample size of $n+1$ for the weight on the edge between nodes i and j corresponding to the rank-k SVD and singular value scaling value of α as

$$t_{ij}^{(1)} = \frac{\bar{e}_{ij}(k,\alpha,n)-e_{ij}(k,\alpha)}{s_{ij}(k,\alpha,n)/\sqrt{n+1}}\ . \tag{6}$$

This one-sample t statistic can be used to identify anomalous, or outlier, edge weights in a single graph. The hypothesis being tested is that there is no difference between an edge weight and its mean value (sampled from weights derived from different LSA models); thus, higher values of the t statistic correspond to more anomalous edge weights.

Similarly, a two-sample t statistic for the corresponding edge weights using SVDs with ranks k_1 and k_2 and sample sizes n_1 and n_2 respectively is defined as

$$t_{ij}^{(2)} = \frac{\bar{e}_{ij}(k_1,\alpha,n_1)-\bar{e}_{ij}(k_2,\alpha,n_2)}{\sqrt{\frac{[s_{ij}(k_1,\alpha,n_1)]^2}{n_1}+\frac{[s_{ij}(k_2,\alpha,n_2)]^2}{n_2}}}\ . \tag{7}$$

This two-sample t statistic is used to identify anomalous edge weights when comparing two graphs. The hypothesis being tested here is that the mean weights of corresponding edges in the two graphs are not different.

As the similarities defined in (2) form the entries of a similarity matrix, E, the t statistics defined in (6) and (7) form the entries of the matrices, $T^{(1)}$ and $T^{(2)}$, respectively. These statistics are viewed for entire graphs using MATRIX VIEWS and DIFFERENCE MATRIX VIEWS, respectively, as defined above.

In the case of sparse similarity matrices, edge weights of 0 are treated as missing values for the statistics defined above, and the sample sizes are adjusted to reflect this.

3.3 You Are Here Views

The YOU ARE HERE VIEWS provide context for both GRAPH VIEWS and MATRIX VIEWS so that the user can keep track of their location within the high-level view as they zoom in to focus on a small region. The red rectangle in the YOU ARE HERE VIEW shows the current view boundaries and position within the larger GRAPH VIEW or MATRIX VIEW. The YOU ARE HERE VIEW is implemented simply as a rectangle drawn over a captured image. Updates to the image are only made when the underlying matrix data being rendered changes; thus, selections are not visible in the YOU ARE HERE VIEW.

Panning or zooming in a graph or matrix view will update the rectangle location and size. Similarly, dragging or scaling the red rectangle will pan or zoom the contents of the graph or matrix view. Often navigating using the YOU ARE HERE VIEW is preferred due to the contextual landmarks this view provides.

3.4 Small Multiples Views

Inspired by Tufte [23], the SMALL MULTIPLES VIEW is a combination of GRAPH VIEWS and MATRIX VIEWS that enables comparisons of up to five different document similarity graphs. The graph views provide a high-level overview of the different clusterings resulting from the parameter changes in the LSA models. Further, each graph view serves as a "label", defining the matrix pairs visualized by the MATRIX VIEWS – specifically, each MATRIX VIEW represents the difference between the graph at the head of its row and the graph at the tail of its column.

Each of the GRAPH VIEWS and MATRIX VIEWS is fully interactive and operates just as any graph or matrix view elsewhere in the application would. Selections made in any one of the views are fully linked to all other views in the application. Zooming and panning permit exploratory navigation within each view. The only limitation is the lack of YOU ARE HERE VIEWS for each graph or matrix, so it is difficult to maintain context. Another difference is that double-click expands the entire SMALL MULTIPLES VIEW, rather than any one view within it.

4 CASE STUDIES

Two case studies are presented in this section, focusing on the problems of rank selection and singular value scaling, respectively.

These studies illustrate how LSAView can be used to interactively determine suitable LSA model parameters for the problem of graph clustering.

4.1 Data

Two sets of data are used in the case studies. The first set, denoted DUC, consists of newswire documents used in the 2003 Document Understanding Conference (DUC) for evaluating summarization systems on clusters of documents [19]. The DUC data is comprised of 298 documents in 30 clusters, with each cluster containing about 10 documents focused on a particular topic or event.

The second set of documents, denoted TECHTC, is from the TechTC-100 Test Collection[1] [9]. Each of the 100 subsets of documents in this collection consists of about 150 HTML documents partitioned into two clusters. The case study results presented here use the `Exp_186330_195558` subset of the TECHTC data, which has the lowest difficulty rating in terms of data clustering.

Note that in the case studies we assume that each data set contains two or more clusters, but we do not use the cluster assignments in selecting a suitable rank to expose the cluster structure or determining a useful scaling of the singular values.

4.2 Rank Selection

Following Shneiderman's Visual Information-Seeking Mantra [21], the process of using LSAView to visually determine the rank of the LSA model most suitable for graph clustering is as follows:

1. Identify a range of potential ranks using the SMALL MULTIPLES VIEW.
2. Choose a rank by comparing graph clusterings using the GRAPH VIEWS, MATRIX VIEWS, and DATA TABLE VIEWS.
3. Validate the chosen rank using the DOCUMENT VIEW.

Note that several iterations may be required for each step.

During step 1 of the rank selection process, coarse steps in rank values can be used to identify changes in the graph clustering of documents over a wide range of LSA model ranks. Figure 2 illustrates this for the DUC data using LSA model ranks of $k = 20, 50, 80, 110, 140$, where the latter is the rank determined to be optimal using a cross validation method [25]. As shown in the figure, though, that "optimal" rank does not reveal any information about the cluster structure in the data, and thus is not useful for the problem of document clustering. The DIFFERENCE MATRIX VIEWS in the figure are colored by differences in edge weights (i.e., document similarities). By visualizing the impact of the rank on both the graph clustering and the changes in the edge weights simultaneously over several LSA models with different ranks, the range of suitable ranks can be narrowed. Specifically, in Figure 2, the LSA model ranks of $k = 20$ and, to a lesser extent, $k = 50$ appear to expose definite cluster structure and have edge weights that are somewhat differentiated from those associated with the other ranks (as depicted by the bold blue and red edge rectangles). Thus, subsequent investigation is focused on ranks closer to those values.

After several iterations of step 1, we arrive at Figure 3, depicting LSA models with ranks of $k = 28, 29, 30, 31, 32$. In this figure, the DIFFERENCE MATRIX VIEWS are now colored by the sample means of the edge weights. As the ranks are very close, these DIFFERENCE MATRIX VIEWS help identify changes in edge weights, as the means of those weights should also be close. Note that the two-sample t statistics could be used in a similar fashion.

The result of Step 2 in the rank selection process is depicted in Figure 4, where GRAPH VIEWS and a DIFFERENCE MATRIX VIEW colored by two-sample t statistics are shown for ranks of $k = 30$ (left) and $k = 32$ (right). The t statistics can be used to quickly identify differences between the two graphs, where a user is easily drawn to the areas with the most significant differences between edges weights (i.e., bold green edge and node rectangles in the DIFFERENCE MATRIX VIEW).

After zoomed inspection of graph clusterings associated with several of the most significant differences found using the views depicted in Figure 4, we arrive at Figure 5, illustrating anomalous links (determined from the t statistics) between document 297 and groups of highly related documents (i.e., linked by bold red edges). Now the similarities of document 297 with other documents differ dramatically for the two LSA models of rank $k = 30$ (left) and $k = 32$ (right). Thus, further inspection of that document is required to help identify the most suitable rank.

This brings us to step 3 of the rank selection process, where manual inspection of the underlying documents is used to validate the selected rank. Figure 6 presents the document view beside the zoomed view of the LSA model of rank $k = 30$. After reading the document and those identified as most similar (i.e., those linked to document 297 in the graphs) for the different LSA model ranks (including rank $k = 32$ and other nearby ranks), we conclude that the rank of $k = 30$ is most suitable. Note here that it is coincidental that

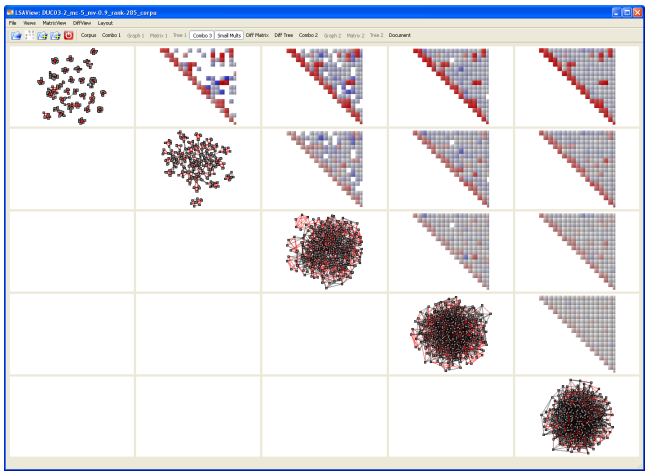

Figure 2: SMALL MULTIPLES VIEW of DUC data with LSA model ranks of $k = 20, 50, 80, 110, 140$. The DIFFERENCE MATRIX VIEWS depict differences in the edge weights across the different graphs.

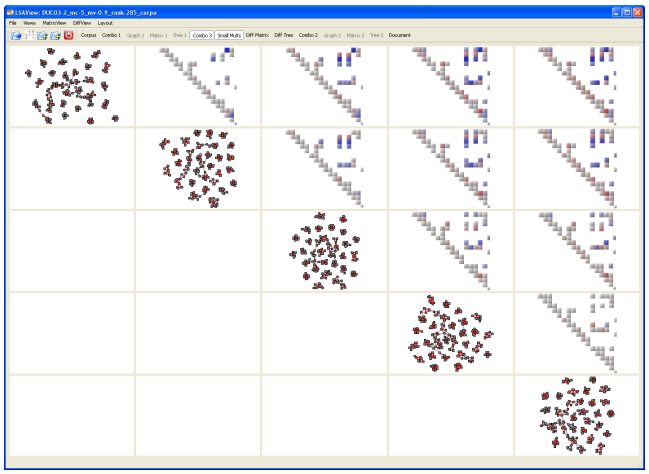

Figure 3: SMALL MULTIPLES VIEW of DUC data with LSA model ranks of $k = 28, \ldots, 32$. The DIFFERENCE MATRIX VIEWS depict differences in the sample means of edge weights between the graphs.

[1] http://techtc.cs.technion.ac.il/techtc100/techtc100.html

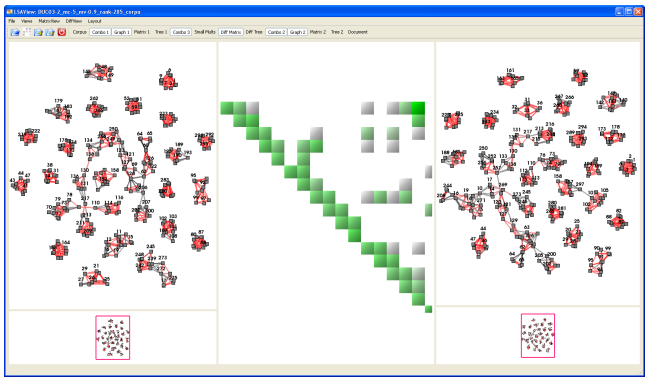

Figure 4: Graph model comparisons of DUC data with rank $k = 30$ (left) and $k = 32$ (right) using the DIFFERENCE MATRIX VIEW (center) of the two-sample t statistics.

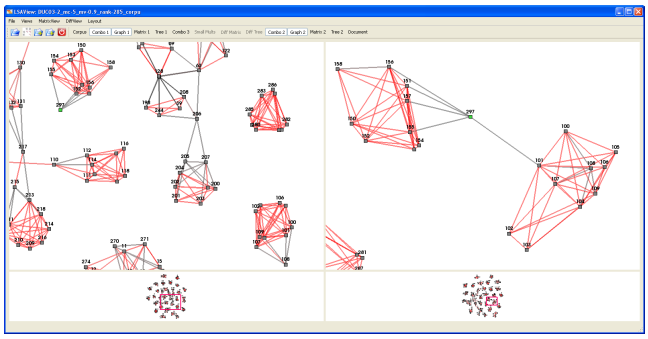

Figure 5: Graph model comparisons of DUC data with rank $k = 30$ (left) and $k = 32$ (right) with weak similarities (gray edges) between node 297 and groups of highly related documents (red edges).

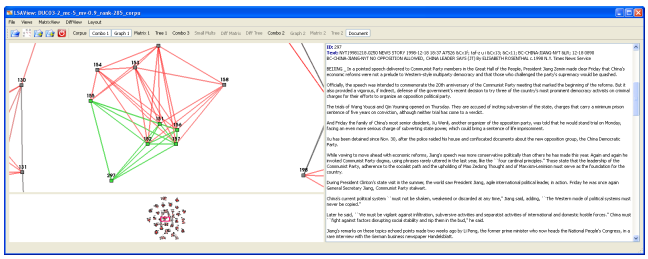

Figure 6: Manual inspection of documents associated with anomalous edge weights is performed using the linked GRAPH VIEWS and DOCUMENT VIEWS. Interacting with both views is necessary for determining where LSA models are linking nodes as expected.

Method	*Rank*	*% Variance*
Leave-one-out cross validation [25]	140	80.72
20-group (fold) cross validation [25]	229	97.27
95% variance [24]	214	95.12
LSAView	30	40.59

Table 1: Rank selection comparison for the DUC data using different methods. Only LSAView exposes the cluster structure in the data.

the most suitable rank is equal to the number of underlying clusters.

Document 297, regarding Chinese leadership statements and policies regarding separatists, turns out to be an anomaly in that it is only tangentially related to the documents in the cluster to which it is assigned - where the main topic is the trial of 3 separatists in China. It appears to be better related to another group of documents, documents 150–158, regarding the policies and responses of the Russian government to Chechnyan separatists. Such subtle relationships would be difficult to assess by simply reading all of the documents in a corpus. By organizing and modeling the data using LSA, combined with interactive exploration of different LSA models, we were able to extract subtle relationships while reading just a few key documents. The strength of this visual algorithm analysis can aid both the developer and analyst in their understanding of the LSA modeling process.

Table 1 shows a comparison of rank selection using LSAView versus cross validation and variance percentage threshold. As mentioned in Section 2.1, cross validation and other existing rank selection methods tend to select ranks which are robust to noise while accounting for variance in the data. However, as shown in Figure 2, using the LSA model with rank determined by cross validation (i.e., $k = 140$), no cluster structure is apparent.

4.3 Singular Value Scaling

Using LSAView to visually determine a singular value scaling for an LSA model that is most suitable for graph clustering follows the same general steps as for the rank selection problem. Again, we use the SMALL MULTIPLES VIEW to determine general trends of the impact of the LSA model parameters—in this case, the singular value scaling parameter, α—followed by more detailed visual analysis using the GRAPH VIEWS and MATRIX VIEWS, with final validation using the DOCUMENT VIEW. The main difference is that it may be useful to first inspect the scaled singular values directly to determine if one scaling may be significantly different. Figure 7 presents the scaled singular values of of the TECHTC data for $\alpha = -2, -1, -0.5, 0, 0.5, 1, 2$. Note that the original singular values are those scaled by $\alpha = 2$. We see that for all of the scaling parameters, the scaled singular values trend toward zero for ranks less than $k \approx 45$. After that, the inverted scalings begin to amplify noise from the data in the LSA model. This indicates that we should concentrate on models with ranks less than $k \approx 45$ or focus on non-inverting scaling parameters.

Figures 8 and 9 present the TECHTC data at ranks $k = 100$ and $k = 20$, respectively, for various singular value scaling parameters ($\alpha = -2, -1, 0, 1, 2$ from left to right along the diagonal). These ranks were chosen as they fell on either side of the value of $k \approx 45$ and thus seemed like a good place to begin comparisons of different LSA models. Since the DIFFERENCE MATRIX VIEWS in these figures depict the differences in edge weights, it is clear in the case of rank $k = 100$ that there is little difference in edge weights between the LSA models using different singular value scaling parameters. However, there are visually apparent differences in the graph clusterings derived from the different models.

Looking more closely at the differences between the LSA models for ranks $k = 100$ and $k = 20$, we find that there are significant differences between the two sets of models. Figure 10 shows zoomed graph views for models at the two ranks using $\alpha = 1$ singular value scaling, i.e., the scaling leading to the best clusters determined visually using the small multiples views. From this figure, we see that the graphs for the rank $k = 100$ model (left) consist mostly of noise, i.e., the links appear random and have very low edge weights. In contrast, the rank $k = 20$ model (right) appears to reflect document relationships within (stronger red links) and between (weaker gray links) clusters. This indicates that models with lower ranks may be more suitable for the graph clustering.

After more comparative analysis between LSA models of different ranks and singular value scaling, we found that models of rank $k = 6$ were most suitable using the criteria presented in the previous section for rank selection. However, determining a suitable singu-

lar value scaling parameter proved more challenging. Again, as for the rank selection problem, we turned to the DOCUMENT VIEW to assist in validating the model. Figure 11 shows the zoomed GRAPH VIEWS for rank $k = 6$ LSA models using $\alpha = 1$ (left) and $\alpha = -1$ (right) singular value scaling parameters. Observe that the model on the left shows two clusters of documents within the larger component (subsets of nodes located close together with high similarity values, i.e., red links) with weak links between them (i.e., light gray links denoting low similarity values). Although the model on the right also shows two distinct clusters in this larger component, there are more strong (red) links between them. As was the case in selecting an appropriate rank, the use of the DOCUMENT VIEW was then used to determine which of the two singular value scalings was more appropriate. For the TECHTC corpus, both scalings appeared appropriate, with the model corresponding to $\alpha = 1$ separating the clusters slightly more than the model with $\alpha = -1$. Indeed both models perform well in separating the clusters, as shown in Figure 12, where the true cluster assignments are encoded by node color.

In the future, we plan to research how to organize and visualize multiple models that lead to equivalent (or nearly equivalent) topological structures in the resulting graphs, as may be the scenario in this latter case of singular value scaling determination.

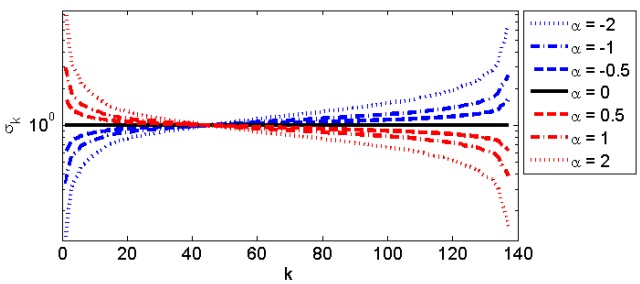

Figure 7: Singular values for the TECHTC data scaled using different values of α. The original singular values correspond to $\alpha = 2$.

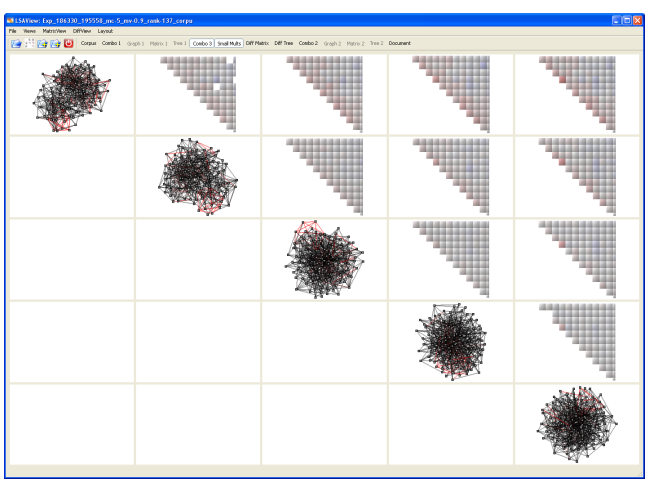

Figure 8: Small multiples view of TECHTC data with LSA model rank of $k = 100$ and singular value scaling parameters $\alpha = 2, 1, 0, -1, -2$ (from left to right along the diagonal). The MATRIX VIEWS depict differences in the edge weights across the different graphs.

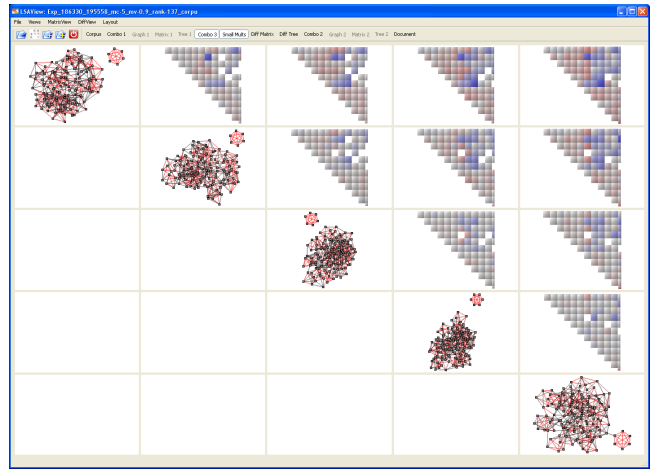

Figure 9: Small multiples view of TECHTC data with LSA model rank of $k = 20$ and singular value scaling parameters $\alpha = 2, 1, 0, -1, -2$ (from left to right along the diagonal). The MATRIX VIEWS depict differences in the edge weights across the different graphs.

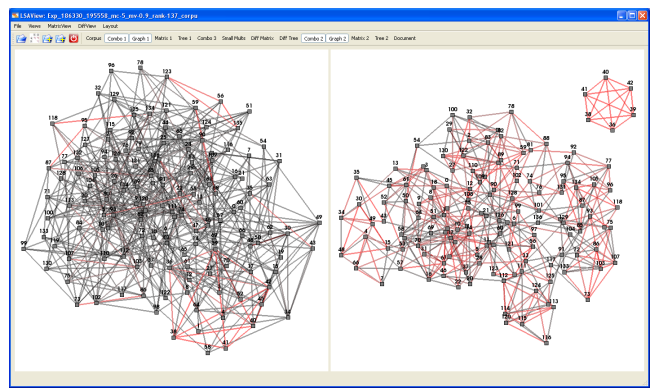

Figure 10: Graph views of TECHTC data for rank $k = 100$ (left) and $k = 20$ (right) LSA models using $\alpha = 1$ singular value scaling.

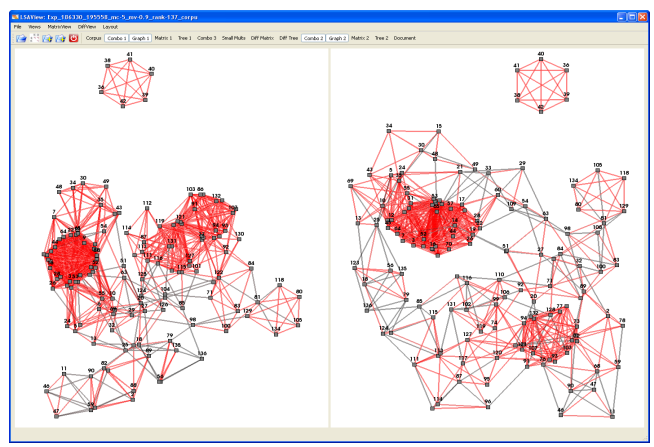

Figure 11: Graph views of TECHTC data for rank $k = 6$ LSA models using $\alpha = 1$ (left) and $\alpha = -1$ (right) singular value scalings.

5 CONCLUSION

In this paper, we have presented the LSAView visual analysis system for interactively assessing LSA models and their impact on

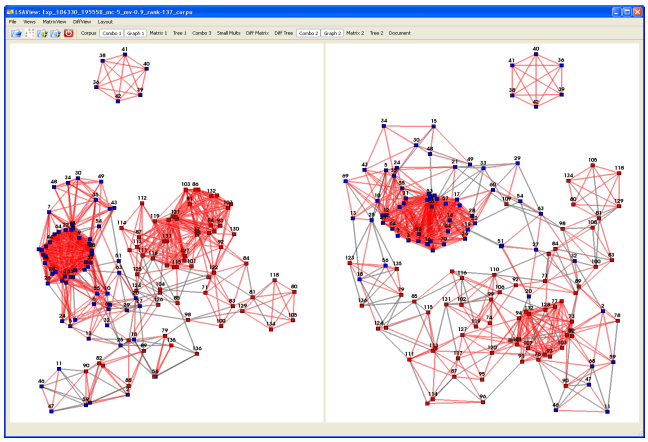

Figure 12: Graph views of TECHTC data for rank $k = 6$ LSA models using $\alpha = 1$ (left) and $\alpha = -1$ (right) singular value scalings and nodes colored by cluster (red for one cluster, blue for the other).

solving text analysis tasks. This is a key departure from previous work in this area, which has tended to focus on algorithm performance rather than what the overall impact would be to the results of the analysis. We have focused on how parameter choices associated with LSA models impact modeling and analysis downstream in complex text analysis pipelines. Through two case studies presented here, we have illustrated how LSAView can be used effectively to understand LSA models, both in terms of how they are used to seed other models (e.g., graph models) and how they can be applied in solving the task of graph-based document clustering.

We have presented several new visualizations for analyzing LSA models, and we have demonstrated how these visualizations can lead to better use of LSA for the problem of document clustering. However, there remain many related open questions, including how well the approach here generalizes to the other variants of LSA, how to identify and visualize sets of LSA model parameters leading to sets of equivalent graphs, and how best to assess performance of visual model exploration systems such as LSAView.

Acknowledgements

This work was funded by the Laboratory Directed Research & Development (LDRD) program at Sandia National Laboratories, a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under Contract DE-AC04-94AL85000.

References

[1] H. Bast and D. Majumdar. Why spectral retrieval works. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 11–18, New York, NY, USA, 2005. ACM Press.

[2] M. Berry and M. Browne. Email surveillance using non-negative matrix factorization. *Computational & Mathematical Organization Theory*, 11(3):249–264, Oct. 2005.

[3] M. W. Berry, S. T. Dumais, and G. W. O'Brien. Using linear algebra for intelligent information retrieval. *SIAM Review*, 37(4):573–595, 1995.

[4] R. B. Bradford. An empirical study of required dimensionality for large-scale latent semantic indexing applications. In *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management*, pages 153–162, New York, NY, USA, 2008. ACM.

[5] P. Crossno, B. Wylie, A. Wilson, J. Greenfield, E. Stanton, T. Shead, L. Ice, K. Moreland, J. Baumes, and B. Geveci. Intelligence analysis using titan. In *Proc. IEEE Symposium on Visual Analytics Science and Technology (VAST)*, pages 241–242, Nov 2007.

[6] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[7] S. T. Dumais, G. W. Furnas, T. K. Landauer, S. Deerwester, and R. Harshman. Using latent semantic analysis to improve access to textual information. In *CHI '88: Proc. SIGCHI Conference on Fuman Factors in Computing Systems*, pages 281–285. ACM Press, 1988.

[8] S. G. Eick, J. Mauger, and A. Ratner. A visualization testbed for analyzing the performance of computational linguistics algorithms. *Information Visualization*, 6(1):64–74, 2007.

[9] E. Gabrilovich and S. Markovitch. Text categorization with many redundant features: Using aggressive feature selection to make SVMs competitive with C4.5. In *Proc. International Conference on Machine Learning (ICML)*, pages 321–328, Alberta, Canada, July 2004. Banff.

[10] D. P. Groth. Visualizing distributions and classification accuracy. In *Proc. International Conference on Information Visualization*, pages 389–394, July 2006.

[11] P. D. Hoff. Model averaging and dimension selection for the singular value decomposition. *Journal of the American Statistical Association*, 102(478):674–685, 2007.

[12] T. Hofmann. Probabilistic latent semantic indexing. In *SIGIR '99: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 50–57. ACM Press, 1999.

[13] T. G. Kolda, B. W. Bader, and J. P. Kenny. Higher-order web link analysis using multilinear algebra. In *ICDM 2005: Proceedings of the 5th IEEE International Conference on Data Mining*, pages 242–249. IEEE Computer Society, 2005.

[14] T. G. Kolda and D. P. O'Leary. A semidiscrete matrix decomposition for latent semantic indexing information retrieval. *ACM Transactions on Information Systems*, 16(4):322–346, October 1998.

[15] T. K. Landauer, D. Laham, and M. Derr. From paragraph to graph: Latent semantic analysis for information visualization. *Proceedings of the National Academy of Sciences of the United States of America*, 101(Suppl 1):5214–5219, 2004.

[16] T. K. Landauer, D. S. Mcnamara, S. Dennis, and W. Kintsch, editors. *Handbook of Latent Semantic Analysis*. Lawrence Erlbaum Associates, 2007.

[17] K. Lerman. Document clustering in reduced dimension vector space. http://www.isi.edu/∼lerman/papers/Lerman99.pdf, 1999.

[18] S. Oba, M. A. Sato, I. Takemasa, M. Monden, K. Matsubara, and S. Ishii. A Bayesian missing value estimation method for gene expression profile data. *Bioinformatics*, 19(16):2088–2096, 2003.

[19] P. Over and J. Yen. An introduction to DUC-2003: Intrinsic evaluation of generic news text summarization systems. In *Proc. DUC 2003 workshop on text summarization*, 2003.

[20] A. B. Owen and P. O. Perry. Bi-cross-validation of the SVD and the non-negative matrix factorization. Technical report, Stanford University, May 2008.

[21] B. Shneiderman. The eyes have it: a task by data type taxonomy for information visualizations. In *Proc. IEEE Symposium on Visual Languages*, pages 336–343, Sep 1996.

[22] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. B. Altman. Missing value estimation methods for DNA microarrays. *Bioinformatics*, 17(6):520–525, 2001.

[23] E. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, 1983.

[24] S. Watanabe and N. Pakvasa. Subspace method in pattern recognition. In *Proc. Int. Joint Conf. on Pattern Recognition*, pages 25–32, 1973.

[25] S. Wold. Cross-validatory estimation of the number of components in factor and principal components models. *Technometrics*, 20(4):397–405, 1978.

[26] B. Wylie and J. Baumes. A unified toolkit for information and scientific visualization. In K. Borner and J. Park, editors, *Proc. Visualization and Data Analysis*, volume 7243, page 72430H. SPIE, 2009.

[27] H. Yan, W. I. Grosky, and F. Fotouhi. Augmenting the power of LSI in text retrieval: Singular value rescaling. *Data Knowledge and Engineering*, 65(1):108–125, 2008.