VisIRR: A Visual Analytics System for Information Retrieval and Recommendation in Large-Scale Document Data

JAEGUL CHOO, Korea University
HANNAH KIM, Georgia Institute of Technology
EDWARD CLARKSON, Georgia Tech Research Institute
ZHICHENG LIU, Adobe Research
CHANGHYUN LEE, Google Inc.
FUXIN LI, Oregon State University
HANSEUNG LEE, Google Inc.
RAMAKRISHNAN KANNAN, Oak Ridge National Laboratory
CHARLES D. STOLPER, Southwestern University
JOHN STASKO, Georgia Institute of Technology
AND HAESUN PARK, Georgia Institute of Technology

In this paper, we present an interactive visual information retrieval and recommendation system, called VisIRR, for large-scale document discovery. VisIRR effectively combines the paradigms of (1) a passive pull through a query processes for retrieval and (2) an active push that recommends items of potential interest to users based on their preferences. Equipped with an efficient dynamic query interface against a large-scale corpus, VisIRR organizes the retrieved documents into high-level topics and visualizes them in a 2D space, representing the relationships among the topics along with their keyword summary. In addition, based on interactive personalized preference feedback with regard to documents, VisIRR provides document recommendations from the entire corpus, which are beyond the retrieved sets. Such recommended documents are visualized in the same space as the retrieved documents, so that users can seamlessly analyze both existing and newly recommended ones. This paper presents novel computational methods, which make these integrated representations and fast interactions possible for a large-scale document corpus. We illustrate how the system works by providing detailed usage scenarios. Additionally, we present preliminary user study results for evaluating the effectiveness of the system.

Categories and Subject Descriptors: H.3.3 [**Information Storage And Retrieval**]: Information Search and Retrieval

General Terms: Design, Algorithms, Performance

Additional Key Words and Phrases: Recommendation, information retrieval, dimension reduction, topic modeling, clustering

1. INTRODUCTION

A deluge of new documents is appearing every day, any one of which might be critical to the questions we are investigating. This presents a challenge, which is similar to looking for a needle in a haystack every day, with limited attention and time resources. This problem is highly under-explored, considering how much efforts have been directed toward developing the related paradigm of web search. Instead, we often have to solve a subtle investigative problem for which each of several documents provides clues. By considering this as an information retrieval (IR) problem, the focus is placed on the long tail, **recall** (making sure that as few relevant documents as possible are missed), while for web search the focus is generally placed on faster gratification of **precision** (making sure that the most relevant documents are contained in the first page of search results).

Author's addresses: J. Choo, Department of Computer Science and Engineering, Korea University; H. Kim and H. Park, School of Computational Science and Engineering, Georgia Institute of Technology; R. Kannan, Oak Ridge National Laboratory; E. Clarkson, Georgia Tech Research Institute; Z. Liu, Adobe Research; C. Lee and H. Lee, Google Inc.; F. Li, School of Electrical Engineering and Computer Science, Oregon State University; C. D. Stolper, Department of Mathematics and Computer Science, Southwestern University; J. Stasko, School of Interactive Computing, Georgia Institute of Technology.

DOI: http://dx.doi.org/10.1145/0000000.0000000

Visual analytics is an effective solution to these high-recall problems. Visual analytics systems for document data can provide an overall understanding about a large set of documents and reveal how the documents are related to each other. Without the help of interactive visualization, this would have been difficult and time-consuming.

Often, exploration of large-scale document analysis involves keyword search. It is a form of "**pull**" technology, in which the user takes actions by forming and issuing queries. However, in the case where high recall is concerned, what queries to issue, for example, with regard to proper keyword usage, becomes crucial in order for users to obtain documents of interest. As a way to compensate for this issue, a **recommendation**, or a "**push**" technology, which the system uses for finding things of interest to recommend to the user, has recently become popular in various study domains. Whereas a search engine is more or less stateless and the same for all users, a recommendation system involves personalization, remembering attributes of the user' interests and search history.

Despite the fact that personalized recommendation seems to be a natural fit to interactive visualization, in the sense of directly utilizing the history of user interactions, there are few examples of such work. To fill this gap, we present, in what we believe to be a milestone study, a novel visual analytics system called VisIRR; an interactive **Vis**ual **I**nformation **R**etrieval and **R**ecommendation for document data, which effectively combines traditional query-based information retrieval with personalized recommendation.

VisIRR utilizes a scatter plot as the main visualization form, similar to IN-SPIRE [Wise et al. 1995]. In other words, topic modeling extracts major topics from a document corpus; documents are then grouped based on their most closely related topics. Afterwards, these documents are projected onto a 2D space via dimension reduction. VisIRR features various novel aspects compared to existing systems; these are described below.

— *Efficient large-scale data processing*. VisIRR currently contains the pre-processed database of half a million documents for various supported computations. Such a database can be efficiently updated with new documents.
— *Interactive visual document analysis via topic modeling, dimension reduction, and alignment techniques*. As core computational modules, VisIRR adopts state-of-the-art methods, nonnegative matrix factorization (NMF) for topic modeling and linear discriminant analysis (LDA) for dimension reduction. They offer a much better quality of results, as well as faster computing time, than traditional methods, including k-means, principal component analysis (PCA), and multidimensional scaling. Additionally, VisIRR supports a novel alignment capability for both topic modeling and dimension reduction, in order to maintain the visualization consistency for easy comparison among different visualization snapshots.
— *Preference-based personalized recommendation*. Given a user's preferences on particular documents during their analysis, VisIRR recommends potentially interesting documents to the users. This recommendation approach enables users to discover documents that cannot be found by imperfect query processes. To perform this recommendation, we developed an efficient PageRank-style graph diffusion algorithm.

In order to integrate all of these capabilities into a sophisticated visual analytics system, we developed various building blocks; from front-end GUI's to back-end computational algorithms. This paper presents these building blocks in detail and with real-world usage scenarios.

The rest of this paper is organized as follows: Section 2 discusses related work; Section 3 describes the user interface design and comprehensive usage scenarios highlighting key capabilities of the proposed system; Section 4 presents our efficient data handling processes using a large-scale data corpus; Section 5 describes the back-end computational methods, which we developed as part of the system. Section 7 is a brief description of the user study, which we conducted for evaluating the system. Finally, Section 8 concludes the paper and discusses future work.

2. RELATED WORK

Information seeking behavior is a complex human activity, which varies dramatically with system capabilities and the user model of these capabilities [Marchionini and Shneiderman 1988]. Ill-defined document search tasks, such as literature search, are often termed 'exploratory search' tasks, in contrast with well-defined tasks such as finding a known, specific item from a set. In the past, traditional information retrieval focused much more on the latter than on the former. More recently, however, advanced approaches were proposed for tackling exploratory search tasks. El-Arini et al. [El-Arini and Guestrin 2011] proposed a new technique, which retrieves relevant documents when given a query of a few documents whose rich meta-data, such as author information, are then utilized in providing recommendations.

In the context of exploratory interfaces, information foraging [Pirolli and Card 1999] and scent theory [Pirolli 1997] suggest making the clusters of related data clear and facilitating the process of finding new clusters of interest. To this end, many systems visualizing the search result also work in concert with automated topic modeling or clustering algorithms, especially when the information space is extremely large or unstructured. IN-SPIRE [Wise et al. 1995] uses the k-means algorithm in order to extract common themes in visualization. iVisClustering [Lee et al. 2012] is an interactive document clustering system focusing on user interactions for improving cluster quality. On the other hand, rather than being restricted to a particular clustering technique, the Testbed system [Choo et al. 2013a] offers users a wide variety of clustering algorithms to choose from and allows the comparison between their results.

Automated recommender systems have often been applied to the problem of matching individual papers from a corpus to individuals from a slate of candidate reviewers [Basu et al. 2001; Wang and Blei 2011]. More relevant to VisIRR are systems that are more exploratory or analytical in nature. The Action Science Explorer (ASE) [Dunne et al. 2012] focuses on co-citation network visualization with document clusters created manually or by heuristics [Newman 2004]. A recently proposed system, called Apolo [Chau et al. 2011], uses a mixed-initiative approach that bootstraps initial user-specified categories and classifications into more comprehensive system-suggested new document categorizations. However, Apolo uses an exemplar-based method where the user is assumed to know a small number of documents within their interest. On the contrary, VisIRR starts from an overview visualization of a fairly large amount of documents retrieved by user queries. Once the documents of interest are identified, VisIRR seamlessly supports an exemplar-based analysis via recommendation processes based on user preference to particular documents; thereby, the user's scope expands gradually beyond the document set retrieved by the initial query.

To our knowledge, even with related work being abundant in this study domain, VisIRR is one of the first systems that *directly consider personalized preference feedback for a large-scale document corpus in an interactive visual environment*.

3. HOW VISIRR WORKS

VisIRR's user interface (UI) is mainly composed of four parts.[1] The *Query Bar* at the top (Fig. 1(A)) enables users to issue queries dynamically, using various fields such as keyword, author name, publication year, and citation count. The *Scatter Plot view* (document details are shown in the lower table) (Fig. 1(B)) visualizes the retrieved documents (as well as the recommended documents) using their topic cluster labels. The color and the size of each node in a scatter plot represent the topic it belongs to and its citation count, respectively. This view can also be generated from any user-selected subset of data (Fig. 1(D)). The *Recommendation view* on the top left (Fig. 1(C)) provides tabular representations of documents rated by users (Fig. 1(C) upper table) as well as of recommended documents (Fig. 1(C) lower table). These recommended documents are also visualized in the *Scatter Plot view* as rectangles; the query-retrieved documents are shown as circles. Finally, the *Label panel* provides additional controls such as highlighting and/or hiding particular topics,

[1]Demo video: https://youtu.be/Dg5oPsZmEjs

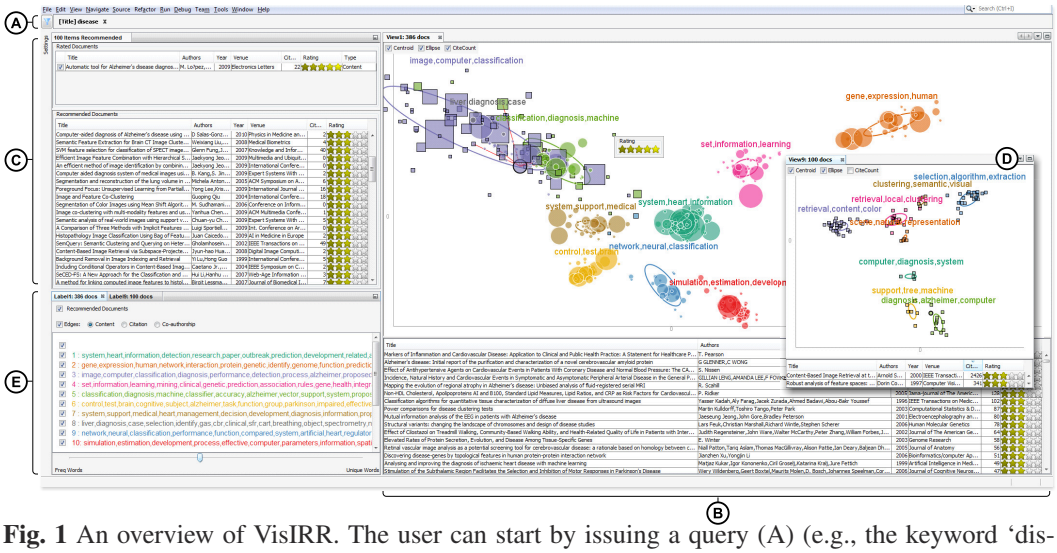


Fig. 1 An overview of VisIRR. The user can start by issuing a query (A) (e.g., the keyword 'disease'). VisIRR visualizes retrieved documents (circles) in a scatter plot and a table view (B) along with a topic cluster summary (E). A node size encodes the citation count. Users can rate documents on a 5-star rating scale in order to indicate their particular interest. Based on preference rating, VisIRR provides a list of recommended items (C), which are also projected back to the existing scatter plot view as rectangles, so that a consistent topical perspective can be maintained. For better understanding, the user can apply *computational zoom-in* on recommended items in order to obtain a much clearer summary (D).

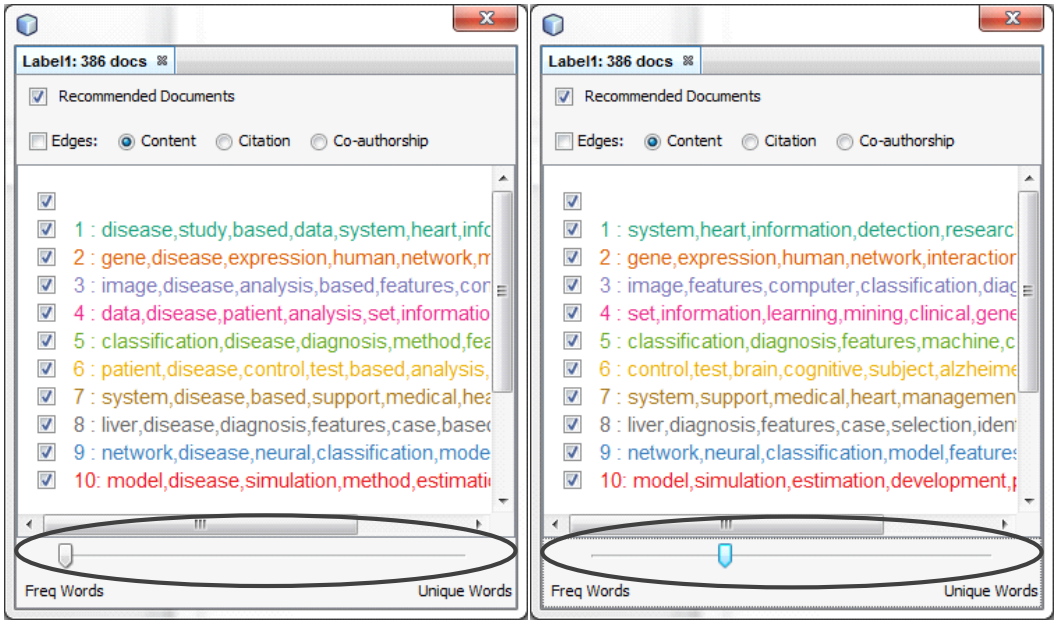
changing how the topic summary labels are chosen, and showing direct edges between the rated and recommended documents (Fig. 1(E)).

3.1. Interactive Visual Document Exploration

VisIRR currently utilizes a publication database called the ArnetMiner data set, which contains approximately 430,000 academic research articles from a variety of disciplines and venues (primarily conferences, journals, and books), as will be described in Section 4. The following scenarios illustrate the utility of VisIRR for tasks related to this data set.

3.1.1. A Visual Overview of Query-Retrieved Documents. In VisIRR, the user starts by issuing queries from the *Query Toolbar*. Suppose the user issues the keyword query 'disease' in a title field; once relevant documents are retrieved based on this query, the system performs topic modeling and dimension reduction steps in order to generate the *Scatter Plot view* (Fig. 1(B)). Since most of the identified topic clusters contain the keyword 'disease,' the user can adjust a slider in the *Label panel* in order to obtain more distinctive words of topic summaries, as shown in Fig. 2. From the *Scatter Plot view*, the user can drill down to a particular topic cluster, such as the topics about 'gene expression data' (top right), and 'image analysis' (top left). By hovering over a cursor, the user can check the document details via a tooltip text and also skim through the document list in the lower table, which is sorted by the number of citations by default. The user can also pan and zoom in order to enlarge a particular topic cluster or an area of interest.

3.1.2. Drilling Down via Computational Zoom-in. The user can drill down a particular topic cluster via our novel interaction called *computational zoom-in*. It enables the user to select an arbitrary subset of documents by visualizing them as a separate view with their own topic modeling and dimension reduction results. For example, the subset may consist of semantically unclear topic



(a) A default topic summary **(b)** A distinct topic summary

Fig. 2 A comparison between default and distinct topic summaries. The query word 'disease' is contained in many topics as one of the most representative keywords (a). Adjusting the slider of *common-vs-unique words* in the *Label panel* improves the distinction between topics (b).

clusters involving multiple topics. On the other hand, the user may select a cluttered region where many points are mixed together.

Fig. 3 shows an example of computational zoom-in interaction. After performing computational zoom-in on a highly cluttered area in the original view, the resulting view successfully reveals clear topics; e.g., 'support vector machines' or 'decision trees,' both of which are widely-adopted techniques in medical image analysis.

3.1.3. Dynamic Queries and Multi-View Alignment. In addition to exploring visualized clusters, the user can apply additional queries in order to further narrow down the retrieved document set. Suppose the user wants to focus on documents published since 2008; then, the user will create another filter from the *Query Toolbar* in conjunction with the previous query in which the keyword 'disease' was used. Given a new set of documents, VisIRR creates another visualization with its own topic modeling and dimension reduction. The user can then compare between the new and previous visualization results, as shown in Figs. 4(a) and (b), respectively, by brushing-and-linking in order to identify, for example, which topic clusters have been either more or less popular since 2008. However, since cluster colors and dimension reduction results are computed independently, it is not possible to easily compare such differences between the two scatter plots.

To solve this problem, VisIRR carries out an alignment step on the new topic modeling and dimension reduction results, with respect to the previous visualization result, so that visual coherence in terms of cluster colors and the spatial coordinates of data points can be maintained. For instance, it is much easier to compare an aligned visualization (Fig 4(c)) against the previous visualization (Fig 4(b)) than it is to compare the unaligned visualization (Fig 4(a)). The aligned visualization helps the user notice that the topic of 'outbreak detection,' shown as a green cluster in the middle of Figs. 4(b) and (c), has not been actively studied since 2008.

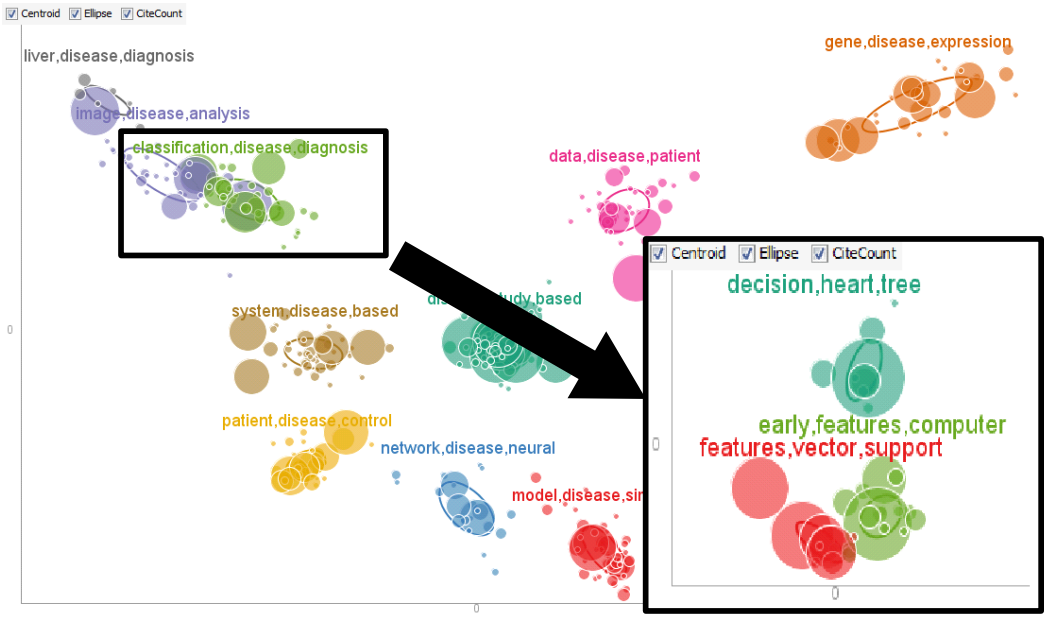
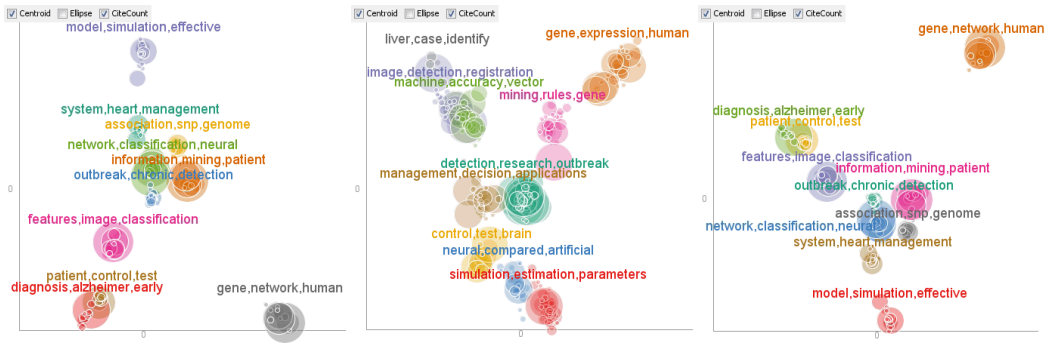


Fig. 3 A *computational zoom-in* interaction. A separate view of user-selected data (a black rectangle at the top left), showing a clear overview of these cluttered data, is created via re-computation of a new topic summary and dimension reduction coordinates.

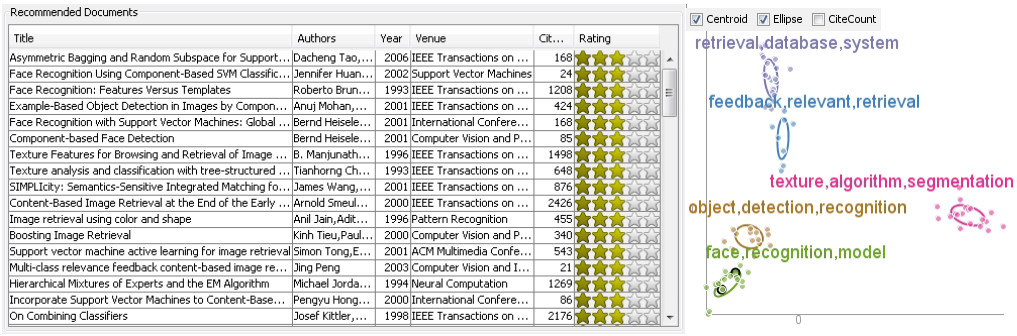


(a) An unaligned view **(b)** A reference view **(c)** An aligned view

Fig. 4 The effects of topic clustering and dimension reduction alignment. A reference view (b) shows documents retrieved by using the query word 'disease' while the other two views (a) and (c) contain their subset published since 2008, with their own topic clustering and dimension reduction steps applied. In an unaligned view (a), it is difficult to compare against the reference view (b) due to the non-correspondence of data point coordinates and topic clusters. However, in the aligned view (c), the topics match those in the reference view (b), in terms of their semantic meanings; thus, their spatial correspondences in the scatter plot are revealed.

3.2. Recommendation

This section describes three types of recommendation capabilities, which are supported by our system through several usage scenarios.



(a) The top-ranked recommended documents

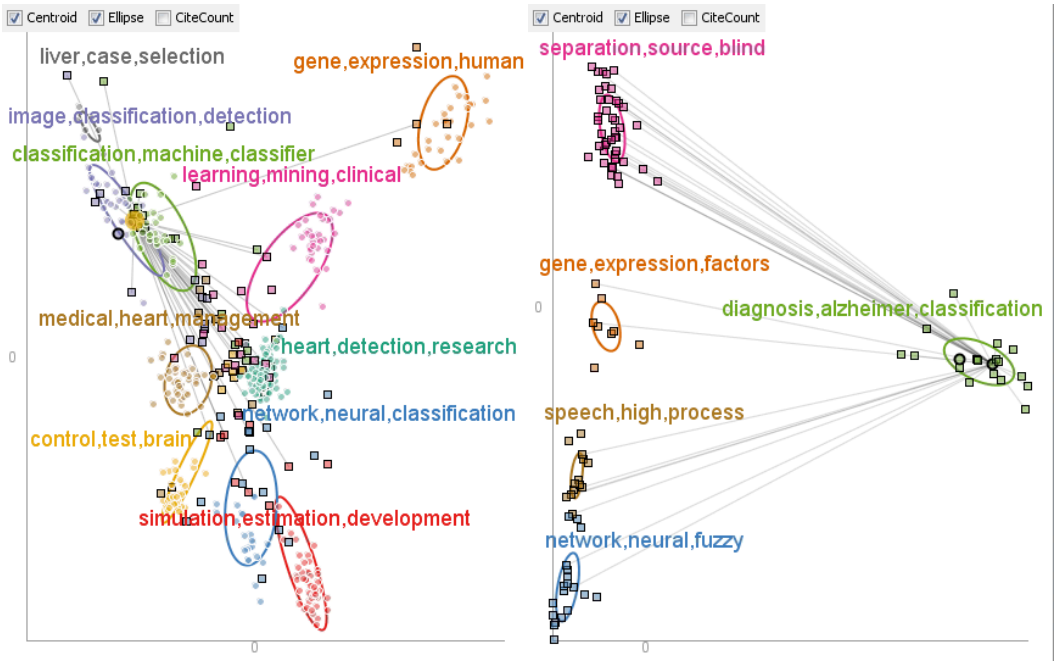
(b) The scatter plot of recommended documents

Fig. 5 Citation-based recommendation results obtained by assigning a 5-star rating to the paper, 'Automatic Classification System for the Diagnosis of Alzheimer Disease Using Component-Based SVM Aggregations.' The relevant papers recommended by VisIRR are mostly papers with high-citation counts.

3.2.1. Content-Based Recommendation. The user can assign ratings to documents to indicate whether she like them or not. Among the retrieved documents, suppose the user found a document titled 'Automatic tool for Alzheimer's disease diagnosis using PCA and Bayesian classification rules' to be interesting and assigned the document a 5-star rating (highly-preferred) by right-clicking the corresponding data point in the scatter plot. Utilizing user preference information, VisIRR discovers the recommended documents based on content similarity. The rated and the recommended documents are displayed in tabular format in the *Recommendation view* (Fig. 1(C)).

From the recommended documents, shown in the lower table (Fig. 1(C)), the user can understand that the research on Alzheimer's disease mainly involves image analysis, clustering, and classification. Notice that without such a recommendation capability provided by VisIRR, the user would not be able to discover these documents since they were not included in the set retrieved by the query. In the scatter plot, the user can see these recommended documents at the upper left corner around the rated document and its adjacent topic clusters. To acquire a better idea about the recommended documents, the user can create another visualization by applying new topic modeling and dimension reduction steps to this subset (Fig. 1(D)). From the new topic summary and visualization, the user can see that the documents directly related to Alzheimer's disease are mainly shown at the bottom half while the upper half of the scatter plot shows documents related to image analysis, such as content-based image retrieval and clustering.

3.2.2. Citation- and Co-Authorship-Based Recommendation. Now, among the recommended documents, the user chooses the document 'Automatic Classification System for the Diagnosis of Alzheimer Disease Using Component-Based SVM Aggregations' and assigns it a 5-star rating. This time, the user changes its recommendation type to citation-based in the *Recommendation view*, in order to obtain highly-cited documents relevant to the rated document. As a result, the top-ranked recommended documents are relatively highly cited papers (Fig. 5(a)). After generating another visualization using only these recommended items, the user can obtain their summary. The items are categorized in topics such as image retrieval, object detection/recognition, face recognition, and texture analysis (Fig. 5(b)). Notice that these types of recommendation results would not be easily obtained by a simple keyword search since the recommended documents do not have specific keywords in common. Instead, they are only implicitly related to the initially chosen document through a citation network, which VisIRR utilizes in order to provide recommendations.

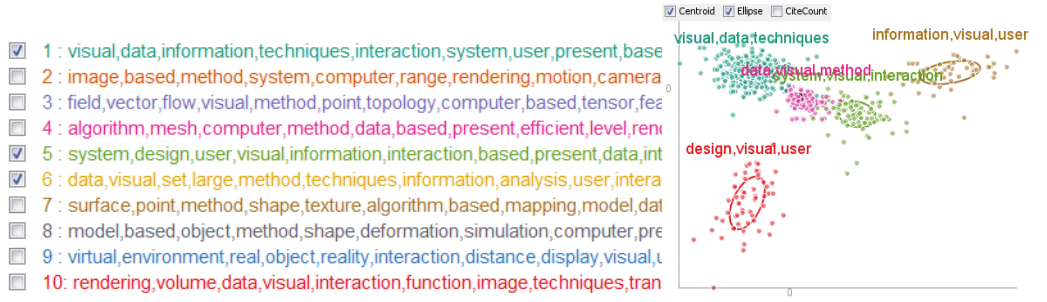


(a) Scatter plot of retrieved and recommended documents **(b)** Scatter plot of recommended documents

Fig. 6 Co-authorship-based recommendation results based on the paper, 'Automatic Classification System for the Diagnosis of Alzheimer Disease Using Component-Based SVM Aggregations.' Edges show direct co-authorship relations from the rated document.

In addition, the user wants to know what other topics or areas of study the authors of the rated paper are involved in. To this end, the user changes the recommendation type to co-authorship-based in the *Recommendation view*. To check direct co-authorship relationships to the rated paper, the user turns on the 'Edges' checkbox by selecting the edge type as 'Co-authorship' in the *Label panel*. The existing visualization of the retrieved documents now includes the recommended documents as well as direct co-authorship relationships of the rated document (Fig. 6(a)). Similar to the previous case, the user can generate another visualization of recommended items in order to acquire a better idea about them. After varying the number of topic clusters, the user obtains a new visualization (Fig. 6(b)). From this new visualization, the user gains knowledge about other studies by the authors of the rated paper, unrelated to Alzheimer's disease (the green topic cluster on the right), in the four areas of blind source separation, gene expression, speech processing, and neural networks. This may potentially indicate that researchers originally interested in Alzheimer's disease diagnosis could expand their research by gaining knowledge on what other domains the authors of the rated paper have published in.

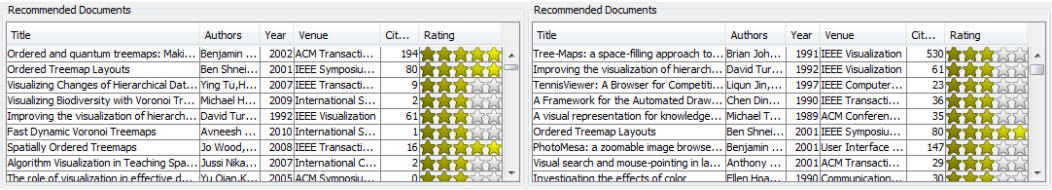
3.2.3. Usage Scenarios. Now, suppose the user wants to use VisIRR in order to find research papers relevant to data visualization. Unsure about what to look for, the user searches for all papers published in those venues whose name contains the keyword 'visualization' in the *Query Toolbar*. Upon examining topics in the *Label panel* and the *Scatter Plot view* (Fig. 7(a)), the user filters out the uninteresting topics of rendering volume/surfaces and performs *computational zoom-in* on the following topics: 'visual, data, information,' 'system, design, user,' and 'data, visual, set.' The user then sees a more detailed topic description (Fig. 7(b)) and starts exploring individual documents. The user selects the paper 'Ordered Treemap Layouts,' which refers to the visualization of hierarchi-



(a) A default topic summary

(b) A new topic summary after *computational zoom-in* on interesting topics

Fig. 7 The initial result based on a keyword query 'visualization' in a venue field. (a) The user selects interesting topics. (b) *Computational zoom-in* reveals more detailed sub-topics after filtering out uninteresting topics in (a).



(a) Content-based recommendation results

(b) Citation-based recommendation results

Fig. 8 Recommendation results obtained by assigning a 5-star rating to 'Ordered Treemap Layouts'

cal data using treemap, and assigns it a 5-star rating. Among the recommended documents, based on content similarity (shown in the *Recommendation view*), the user also rates two documents,'Ordered and Quantum Treemaps: Making Effective Use of 2D Space to Display Hierarchies' and 'Spatially Ordered Treemaps,' as 5-star (Fig. 8(a)). The user notices that the three papers, which she rated, are all recently published papers on the topic of treemap visualization. Now, the user wishes to find representative papers on the topic of the treemap visualization technique, for the purpose of citing them in her paper. The user uses citation-based recommendation, with regard to the rated papers, and finds 'Tree-Maps: A Space-Filling Approach to the Visualization of Hierarchical Information Structures' on the top of the recommendation list (Fig. 8(b)). This is the first paper to propose the technique and also the most popular paper on this specific subject.

4. DATA COLLECTION / INGESTION

In this section, we present the data collection and processing, which we performed, in more detail.

4.1. Initial Data Collection

VisIRR can handle a large-scale document corpus with a rich set of features efficiently. To this end, we started with the ArnetMiner data set, which is composed of approximately half a million academic papers, books, etc. [Tang et al. 2008].[2] The original data set has numerous missing values and inconsistencies with regard to author name, publication venue, etc. To clean-up the data, we utilized Microsoft Academic Search API[3] in order to obtain the full information about the document;

[2]The used data is available as 'DBLP-Citation-network V5' at http://arnetminer.org/citation.
[3]http://academic.research.microsoft.com/About/Help.htm.

thereby, the missing values were filled and the inconsistencies were fixed. VisIRR currently contains 432,605 documents spanning from year 1825 to 2011.

4.2. Data Ingestion

VisIRR maintains the data information in three different forms: (1) original fields of data, (2) vector representation, and (3) graph representations. These are maintained in an efficient and scalable manner. To efficiently manage the large amount of data in all of these forms, we optimized various data processing/storage techniques through database construction, pre-computation of frequently used information, and balanced storage between disk and memory, which will be described in more detail below.

4.2.1. Original Data Attributes. For efficient and flexible query support, we encoded the original data into an SQL database including full-text search capabilities on title, keywords, abstract, and venue fields. To perform topic modeling and dimension reduction steps, we pre-computed the sparse vector representations of individual documents by integrating title, keywords, and abstract fields using the bag-of-words encoding scheme. Each vector representation is stored in a single file on a hard disk drive where the name of the file is the document ID. In this manner, VisIRR is able to retrieve the vector representations of documents using their document ID's in the time complexity of $O(1)$.

4.2.2. Vector Representation. VisIRR manages the vector representations of documents in a similar manner to cache replacement algorithms; that is, the vector representations already loaded into the memory are referenced from the memory and will be reloaded from a disk when they are needed again. When the amount of memory-loaded vectors exceeds a pre-defined limit, the least recently used vectors are removed from memory. When needed later, they are loaded from a disk once again. In this manner, we avoid loading the entire vector representation of all the documents from the beginning, which would consume a significant amount of time and memory at system startup. VisIRR also prevents memory usage from blowing up due to the long-term usage of the system.

4.2.3. Graph Representation. The recommendation module (Section 5) requires an input graph where nodes correspond to documents and edges represent their pairwise similarities/relationships. We pre-computed three such graphs for the entire data set using contents, citation, and co-authorship, respectively, for the purpose of supporting diverse recommendation capabilities. For a content-based graph, we computed the pairwise cosine similarities between document vectors. Since the pairwise information requires $O(n^2)$ storage where n is the total number of documents, we maintain the partial information regarding each document's similarity to ten most similar documents. For a citation graph, edges are formed between document pairs if one had cited the other. For the co-authorship graph, edges are created if two documents share the same author(s). For each graph, VisIRR maintains the data structure about each document's list of edges, in terms of the destination document and its edge value so that it can retrieve the edge information for particular documents in $O(1)$ time complexity.

4.3. Scalable Update for New Data

It is crucial to have the capability of efficiently expanding the stored information for newly added documents. This task involves obtaining the representations of new documents as well as updating information about existing documents. For instance, when updating the content similarity graph, where the ten most similar documents and their cosine similarity values are kept, we have to compute the pairwise similarity between all the existing documents and all the new documents. Then, we have to compare these similarity values against the current top ten similarity values and replace them accordingly. The process is done in $O(n \times n_{new})$ time complexity where n and n_{new} are the numbers of existing and new documents, respectively.

5. COMPUTATIONAL METHODS

In VisIRR, we developed various computational modules based on topic modeling, dimension reduction, alignment, and graph-based recommendation, each of which is described below.

5.1. Topic Modeling

Topic modeling provides a summary of a given set of documents in terms of its major topics. The resulting topic indices are used to color-code documents in a scatter plot with each topic's representative keywords (Figs. 1(B) and (E)). Traditionally, topic modeling approaches based on probabilistic graphical modeling, such as probabilistic latent semantic indexing [Hofmann 1999] and latent Dirichlet allocation [Blei et al. 2003], have been widely used. In our study, however, we employed a technique which recently became popular; it is called nonnegative matrix factorization (NMF) [Kim and Park 2007] due to its output consistency from random initialization and also due to its computational efficiency [Choo et al. 2013b]. In addition, NMF has exhibited superior performance in document clustering compared to traditional methods such as k-means [Kim and Park 2008; Xu et al. 2003], which was used in IN-SPIRE [Wise et al. 1995].

Given a nonnegative matrix $X \in \mathbb{R}^{m \times n}$, and an integer $k \ll \min(m, n)$, NMF finds a lower-rank approximation given by

$$X \approx WH, \tag{1}$$

where $W \in \mathbb{R}^{m \times k}$ and $H \in \mathbb{R}^{k \times n}$ are nonnegative factors. In the context of topic modeling and document clustering, each column vector $x_i \in \mathbb{R}^{m \times 1}$ of X represents an individual document as an m-dimensional vector using bag-of-words encoding, along with additional pre-processing steps such as inverse-document frequency weighting and L_2-norm normalization. The value of k represents the number of topics; each column of W represents a topic, where the value of a particular dimension indicates the weight of the corresponding keyword in the topic. By choosing keywords with the most significant weight values, we obtain their topic summary. To perform document clustering, we utilize each column of H as the soft clustering vector representation of a document such that the column vector $h_i \in \mathbb{R}^{k \times 1}$ of H represents a soft clustering vector for the i-th document, and the cluster index of the document can be obtained as the dimension index with the largest value in h_i.

The particular NMF algorithm used in our study was based on a recently proposed block principal pivoting algorithm [Kim and Park 2011],[4] which is one of the fastest, numerically stable algorithms. In Section 6, we present the quantitative evaluation, which shows the advantage of NMF in topic modeling applications.

5.1.1. Computational Complexity. The overall computational complexity of NMF is difficult to determine since it goes through iterative updates of W and H in a block-coordinate descent framework until its convergence to a local minimum. The dominant computation for updating each W and H takes $O(mnk)$ for iterations until we find an optimal passive index set for the nonnegativity constraint.

5.2. Dimension Reduction

Dimension reduction computes 2D representations of documents in a scatter plot (Fig. 1(B)). VisIRR adopts a supervised dimension reduction method called linear discriminant analysis (LDA) [Howland and Park 2004], which, unlike traditional methods such as PCA and MDS, explicitly utilizes additional cluster label information taken from the document clustering results described above. Using this information, LDA tries to highlight the cluster structure in low-dimensional space.

In detail, given a data matrix $X \in \mathbb{R}^{m \times n}$, whose column vectors $x_i \in \mathbb{R}^m$ for $i \in \{1, 2, \cdots, n\}$ represent data items and their cluster labels l_i, LDA first computes the high-dimensional statistic

[4]The source code is available at http://www.cc.gatech.edu/~hpark/nmfsoftware.php.

called the within- and the between-scatter matrices, S_w and S_b, respectively, as,

$$S_w = \sum_{i=1}^{n} (x_i - c_{l_i})(x_i - c_{l_i})^T \text{ and}$$

$$S_b = \sum_{i=1}^{n} (c_{l_i} - c)(c_{l_i} - c)^T,$$

where c_{l_i} and c represent the centroid of cluster l_i for $l_i \in \{1, 2, \cdots, k\}$ and the global centroid, respectively, i.e.,

$$c_{l_i} = \sum_{j=1}^{n} x_j I(l_j = l_i) / \sum_{j=1}^{n} I(l_j = l_i) \text{ and}$$

$$c = \frac{1}{n} \sum_{i=1}^{n} x_i.$$

Once S_w and S_b are computed, LDA obtains its linear transformation matrix of LDA, $G_{LDA} \in \mathbb{R}^{2\times m}$, which maps an m-dimensional data vector x to a two-dimensional vector $z = Gx$, by solving

$$G_{LDA} = \arg \max_{G \in \mathbb{R}^{2\times m}} \text{trace}\left(\left(GS_w G^T \right)^{-1} \left(GS_b G^T \right) \right). \quad (2)$$

The columns of G_{LDA}, which is the optimal solution to this equation, are obtained as the leading generalized eigenvectors v of the following generalized eigenvalue problem [Fukunaga 1990]

$$S_b v = \lambda S_w v.$$

To ensure numerical stability of the matrix inverse in Eq. (2), VisIRR uses a regularized version of LDA, which replaces S_w by $S_w + \gamma I$. In practice, the parameter γ controls how compactly LDA represents each cluster in a 2D scatter plot. We provide a slider interface for changing the value of γ, enabling users to focus their analyses at either a cluster level or an individual document level. For more details, we refer readers to [Choo et al. 2009, 2010].

5.2.1. Computational Complexity. The computational complexity of LDA is mainly governed by the generalized eigenvalue problem. By applying QR decomposition on a data matrix X, we can solve this problem, whose computational complexity is $O\left(mn^2\right)$ [Park et al. 2007].

5.3. Alignment

In VisIRR, users can dynamically create multiple scatter plots with (1) different parameter values, e.g., the number of topic clusters in NMF and a regularization in LDA, and (2) a new set of data from a different query or user selection. In order to maintain consistency between different scatter plots and facilitate their easy comparison, VisIRR aligns different topic clustering and dimension reduction results. By aligning the topic clustering results, the same topic cluster indices are expected to have coherent meanings. By aligning dimension reduction results, the same data points are located in a similar position within a 2D space between different scatter plots.

For topic cluster alignment, VisIRR utilizes the Hungarian algorithm [Kuhn 1955]. Given two sets of cluster assignments for data items, the Hungarian algorithm finds the optimal matching of cluster indices between the two sets so that the number of common data items within the matching cluster pairs can be maximized. Based on the matching result, VisIRR changes the topic cluster indices and the colors of the newly created scatter plot with respect to those in the reference scatter plot. In this manner, VisIRR maintains the topic cluster indices/colors with their consistent semantic meanings among multiple visualization results.

For the alignment of dimension reduction results, we employ the technique called Procrustes analysis [Hurley and Cattell 1962; Eldén and Park 1999], which finds the best mapping from one result to the other via high-dimensional rotation. Procrustes analysis has been widely applied to

image registration in the field of computer vision; however, it has never been used in interactive visualization applications. Furthermore, we improved the original Procrustes analysis by incorporating translation and isotropic scaling factors; that is, given two reduced-dimensional matrices in a two-dimensional space, $X, Y \in \mathbb{R}^{2\times n}$, where n represents the number of data points, respectively, our alignment algorithm solves

$$\min_{Q,\mu_X,\mu_Y,k} \left\| \left(X - \mu_X 1_n^T\right) - kQ\left(Y - \mu_Y 1_n^T\right) \right\|_F, \tag{3}$$

where $Q \in \mathbb{R}^{2\times 2}$ is an orthogonal matrix (performing the rotation in a two-dimensional space), μ_X and μ_Y are two-dimensional column vectors (performing translation), k is a scalar (performing isotropic scaling), and 1_n is an n-dimensional column vector whose elements are all 1's. The solution for Eq. (3) can be obtained as follows. First, we perform a singular value decomposition on $\left(Y - \mu_Y 1_n^T\right)\left(X - \mu_X 1_n^T\right)^T$ as

$$\left(Y - \mu_Y 1_n^T\right)\left(X - \mu_X 1_n^T\right)^T = U\Sigma V^T,$$

where $U, V \in \mathbb{R}^{2\times 2}$ are orthogonal matrices and $\Sigma \in \mathbb{R}^{2\times 2}$ is a diagonal matrix. Now, the optimal solutions of Q, and k are obtained as

$$Q = VU^T, k = \text{trace}(\Sigma)/\text{trace}\left(\left(Y - \mu_Y 1_n^T\right)\left(Y - \mu_Y 1_n^T\right)^T\right),$$

and μ_X and μ_Y are obtained as the column-wise mean vectors of X and Y, respectively.

This alignment step helps users understand how similar or different the placement of the corresponding data items and topic clusters are between different views.

5.3.1. Computational Complexity. The $k \times k$ co-membership frequency matrix between the two sets of k clusters is the input to the Hungarian algorithm. It takes $O(n)$ computations, where k is the number of clusters and n represents the number of data items. Then the Hungarian algorithm has the computational complexity of $O(k^3)$. The input to Procrustes analysis also takes $O(n)$ computations; then, the main algorithm runs efficiently since it works on the matrix of size 2×2. As will be seen in Section (6), these alignment algorithms have minimal effects on the running time of VisIRR.

5.4. Recommendation

The main input to the recommendation algorithm is the personalized preference to particular documents, which are interactively assigned by users on a 5 star rating scale (Fig. 1(B)). All the documents are initially set to have a 3-star rating (neutral preference); however, users can interactively assign ratings to documents, where 1 star corresponds to a preference value of -2, and 5 stars correspond to +2, etc.

Given such a user preference input, VisIRR identifies documents to recommend by performing a graph diffusion algorithm on a weighted graph of the entire document corpus. Such a graph can be based on contents, citation network, or co-authorship network, depending on the user's choice (Section 3.2). In particular, VisIRR adopts a heat kernel-based graph diffusion algorithm [Chung 2007], which gives much faster convergence than traditional algorithms. In detail, given an input graph $W \in \mathbb{R}^{N\times N}$ between N documents, where each column of W is normalized to have a unit L_1-norm, and a user preference vector $p \in \mathbb{R}^{N\times 1}$, where the i-th component p_i indicates the preference value of the i-th document, VisIRR computes the recommendation score vector $r \in \mathbb{R}^{N\times 1}$ of N documents as

$$r = \alpha \sum_{k=0}^{n} (1-\alpha)^k W^k p, \tag{4}$$

where α and n are user-specified parameters currently set to $\alpha = 0.7$ and $n = 3$. An intuitive explanation of this formulation is that the preference value p is propagated to its neighboring nodes with the corresponding weights specified in graph W during the first iteration. Then, the resulting values

are propagated again on the same graph W with the scaling $(1-\alpha)$ at the next iteration, and so on. Finally, those values computed from each iteration are added up to form the final recommendation score vector r. Once converged, VisIRR selects the documents with the biggest scores in r as the documents to recommend.

All the computations in this algorithm, which are basically matrix-vector multiplications, are performed based on sparse representations. Therefore, as long as W and p have a small number of non-zero entries, the computation is usually fast. In addition, VisIRR supports the capabilities of interactively adding/removing the rated documents as well as changing the ratings of the existing documents. Such computations are performed dynamically per interaction, which essentially makes p have only one non-zero entry. It allows us to maintain the real-time efficiency of computations during frequent user interaction.

5.4.1. Computational Complexity. The running time of the recommendation algorithm varies significantly. Thus, it is difficult to define its computational complexity since it would depend on the number of seed documents' edges, as well as on those of their neighbors. Usually, if the seed documents have the smaller number of neighbors, the recommendation algorithm would run faster.

5.5. Implementation

The front-end UI and visualization of the system were implemented in JAVA, partly based on the FODAVA testbed system [Choo et al. 2013a]. NetBeans Rich Client Platform and IDE[5] were used for flexible window management. The back-end computational modules, NMF and LDA, were originally written in MATLAB, but were later converted into a JAVA library.[6] For querying and accessing the database, we used the H2 library.[7]

6. QUANTITATIVE EVALUATION

In this section, we present the quantitative evaluation results. First, we explain our choice of the topic modeling module's design. Next, we report the running time of each module.

6.1. Comparison of Topic Modeling Methods

We experiment with four well-known document clustering and topic modeling methods: NMF,[8] LDA,[9] k-means,[10] and information bottleneck (IB) [Slonim and Tishby 2000].[11] For k-means and IB, which do not explicitly produce topics but rather give document clusters, we treated each cluster as a topic. For k-means, in order to obtain its representative keywords, each centroid was regarded as the word distribution vector of its corresponding topic. For IB, from each word cluster, we selected the keywords with the highest probability values over those documents belonging to the corresponding topic.

Among the four methods, we compared the computing time and topic coherence score. To evaluate topic coherence, we used pointwise mutual information (PMI) [Newman et al. 2010], which, when given the top keywords $w_1, w_2, \ldots, w_t$ of a topic, was computed as

$$PMI = \sum_{i=1}^{t-1} \sum_{j=i+1}^{t} \left(\log \frac{p(w_i, w_j)}{p(w_i)\, p(w_j)} \right),$$

[5]http://netbeans.org/features/platform/index.html
[6]http://www.mathworks.com/products/javabuilder/
[7]http://www.h2database.com/html/main.html
[8]We used code available at https://github.com/kimjingu/nonnegfac-matlab.
[9]We used code available at http://psiexp.ss.uci.edu/research/programs_data/toolbox.htm.
[10]We used built-in MATLAB function.
[11]We used code available at http://ai.stanford.edu/~gal/Code/ibsi_sequ.m.

Table I The PMI scores averaged over 20 runs. The best performance values are shown in bold.

Filtering keywords	Data size (term×doc)	**NMF**	**LDA**	***k*-means**	**IB**
'lagrange'	3,947×515	**0.8535**	0.7015	0.5874	0.6709
'disease'	12,515×2,608	**0.7426**	0.6918	0.3812	0.4483
'scalable'	25,664×13,112	0.5314	**0.6203**	0.2904	0.3412

Table II The computing times averaged over 20 runs. The best performance values are shown in bold.

Filtering keywords	Data size (term×doc)	**NMF**	**LDA**	***k*-means**	**IB**
'lagrange'	3,947×515	6.62	5.50	**5.12**	26.32
'disease'	12,515×2,608	**28.82**	36.68	249.93	113.74
'scalable'	25,664×13,112	**134.55**	231.44	2,651.10	265.76

Table III Computing times of each module averaged over three runs. Standard deviation is included in parenthesis for the recommendation module, which showed high volatility.

	2,000 docs	971 docs	515 docs
Topic modeling	12,152	7,244	3,953
Dimension reduction	8,995	4,070	1,903
Alignment	1,097	867	899
Recommendation	605 (452)	989 (585)	1,330 (692)

where $p(w_i, w_j)$ represents the probability of w_i and w_j co-occurring in a common document and $p(w_i, w_j)$ is the probability of w_i occurring in a document. We used the top ten keywords, i.e., $t = 10$. For each model, we report the averaged PMI score over all calculated topics.

We set the number of topic clusters to 20. For the other parameters, we used the default setting available in the original implementation. All implementations were developed in MATLAB. For the data sets used in this part of the study, we filtered the ArnetMiner data set mentioned in Section 4 using a particular keyword to generate data subsets for the experiment.

Table I shows the PMI scores averaged over 20 runs. Overall, NMF and LDA show higher topic coherence compared to *k*-means and IB. In terms of the computing times in Table II, NMF was shown to run the fastest for large data sets. These experimental results demonstrate the superiority of NMF, which was used as the core topic modeling module in VisIRR.

6.2. Running Time Breakdown

Table III shows the computing times of our computation modules averaged over three runs for data sets of different sizes. Topic modeling and dimension reduction modules take less time when the data set size is smaller. On the other hand, the time consumed by alignment stays relatively the same with various data set sizes. The running time of the recommendation module shows high volatility, in the range from 50 seconds to 2,000 seconds per iteration, since it is mostly affected by the rated documents' edge count, as discussed in Section 5.4.1.

7. CONFIRMATORY USER STUDY

It is acknowledged that evaluation of information visualization and visual analytic systems are challenging [Plaisant 2004]. Insight-based evaluation [Saraiya et al. 2005; Plaisant et al. 2008] has recently gained popularity as an alternative to traditional time-and-accuracy measures. As a preliminary gauge of how well our usage scenarios matches real user behavior, we conducted an end-user evaluation of VisIRR.

Table IV The UI action counts across all participants and tasks.

Action	Description	Count
Checking tooltip text	A tooltip showing document details triggered by hovering over a table row or a scatter plot node	38,897
Rating documents	The user assigns a non-default 1-5 star rating from table entries or scatter plot nodes	80
Checking document details	The user opens the detail dialog box for one or more documents	146
Bookmarking documents	The user copies document information to the clipboard	35
Performing filtering	The user performs a filter (by keyword, year, citation count, or author's name) on the current results	24

This study has been designed to provide evidence-by-existence; that is, our goal was to provide support for our implicit VisIRR design claims. For example, we sought to show that recommendations outside the initial query-retrieved documents are helpful in finding useful documents and that VisIRR serves its intended purpose when utilized by real users. This should prove that our assumptions made in the user scenarios discussed in Section 3 were valid.

7.1. Procedure

The participants in the study were first provided with a live demo of the system (lasting five to ten minutes, depending on questions). Then, the participants used the system to conduct searches using their own queries and to complete a set of pre-defined tasks in the field of either *ubiquitous computing* or *information visualization* (e.g., "Describe any apparent subfields or application areas of information visualization."). Finally, we deployed a version of the IBM Computer System Usability Questionnaire (CSUQ) [Lewis 1995] along with a few other subjective assessment questions specific to VisIRR.

The system was deployed on a workstation with 2.5GHz Intel Xeon processors and 8 GB RAM running 64-bit Windows 7. The workstation included a 30-inch monitor for VisIRR and a 19-inch monitor (as a task response window).

We recruited seven male PhD students between the ages of 24-40 enrolled in various technical degree programs (engineering, computer science, and robotics). As such, they were all experienced in researching academic literature using online resources such as Google Scholar and the IEEE/ACM digital libraries. We asked the participants to self-rate their familiarity with information visualization and ubiquitous computing literature; all the participants self-rated four or less on a seven-point Likert scale for information visualization; six out of seven students did the same for ubiquitous computing. Participants completed the tasks with regard to the area they were less familiar with. VisIRR was configured to log the user's UI actions; their action types are summarized in Table IV. We observed users non-intrusively while they completed tasks.

7.2. Results

Table IV shows the raw action count across all users and all tasks. Although we do not provide rigorous comparison against other baseline settings, these counts partially support our subjective impression, which was formed while watching users complete tasks; the users consistently made use of major VisIRR features (visualization, ratings, recommendations, and details-on-demand). Since one of our most basic questions was whether the users would actually make use of the novel features such as ratings and recommendations, this preliminary result was encouraging. The numbers of checking tooltip text in Table IV are somewhat exaggerated because VisIRR tooltips have a very short timeout triggering their appearance.

All the users made at least nine distinct document ratings across all tasks; interestingly, they did so relatively evenly, from different portions of the UI (the recommended documents, the query lists,

and the scatter plot). Document details were disproportionately triggered from the visualization (112/146), indicating that the participants both interacted with the visualization and drilled down into document details from there. Although this may have been due to the relatively small panel size assigned to the query-retrieved document list, it confirmed both our subjective observations and post-test user comments, e.g., "*It's good to have that first clustering result ... It's easy to go deeper down from one or two clusters.*"

On the subjective CSUQ, scores were generally five or higher, with the lowest rated scores observed for questions, such as "*The system has all the functions and capabilities I expect it to have*"; "*The system gives error messages that clearly tell me how to fix problems*"; and *"Whenever I make a mistake using the system, I (am able to) recover easily and quickly*." We suspect that these ratings reflect occasional software bugs and crashes, which occurred during some of the participant sessions.

Our results also suggest a potential interesting contrast in user behavior with more traditional keyword-based search algorithms; in exploratory tasks with keyword search engines, one might expect to see multiple iterations of keyword refinement and to inspect results for a given task; however, our users performed relatively few filter actions (all keyword refinements rather than by author, time, or citation). Because VisIRR recommendations expand the search query outside its original bounds (and highlight those nodes outside the bounds), iterating keyword terms is less necessary. Of course, we hypothesize that rating-based refinement is more productive, since it requires less expertise from the user in generating useful keyword sequences; at least one user clearly agreed by saying that VisIRR "*... is definitely much better than blindly searching (on) Google Scholar or (on) basic search engines using just a few keywords.*"

8. CONCLUSION AND FUTURE WORK

In this paper, we presented a visual analytics system called VisIRR, which is an interactive visual information retrieval and recommendation system for document discovery. One of the primary contributions of VisIRR is that it effectively combines both paradigms of passive query processes and active recommendation by reflecting user preference feedback. In addition, VisIRR tackles a large-scale document corpus directly, through efficient data management and by updating new data, as well as through a suite of state-of-the-art computational methods such as topic modeling (e.g., NMF), dimension reduction (e.g., LDA), alignment (e.g., Hungarian algorithm and Procrustes analysis), and personalized recommendation (e.g., heat kernel-based graph diffusion algorithm)

In future work, we plan to support efficient, interactive topic modeling and 2D layout algorithms. In fact, many users have often mentioned visualization not coming up immediately due to the non-trivial computational time of the various algorithms involved. To this end, the development of parallel and distributed algorithms can improve the usability of the system, in terms of responsivity and speed.

Additionally, users sometimes tried to move documents or clusters to see what other documents or clusters move correspondingly. Fast and interactive topic modeling and layout algorithms, which incorporate such user feedback, would substantially improve the usability of VisIRR [Endert et al. 2012].

Moreover, VisIRR's recommendation module relies heavily on citation and co-authorship network information, which may not be readily available for relatively new documents. To solve this issue, we plan to integrate additional approaches in order to extract structured information, such as entity resolution and disambiguation, which are properly distinguished among different authors with the same name.

Finally, we plan to expand the capabilities of VisIRR to other types of document data analysis, such as social media data and news articles. These types of data, however, poses other challenges; for instance, citation or co-authorship information may not be available. Furthermore, topic modeling results may not be reliable given documents with a short length. These issues can limit VisIRR's recommendation capabilities; to handle them, additional information, such as social network and/or co-viewing information, should be utilized when making a recommendation. In addi-

tion, other topic modeling and document embedding methods, suitable for short documents, could be used in VisIRR [Yan et al. 2013; El-Arini et al. 2013].

ACKNOWLEDGMENTS

The work of these authors was supported in part by the National Science Foundation grant CCF-0808863 and and Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (NRF-2016M3C1B6950000).

This manuscript has also been co-authored by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725 with the U.S. Department of Energy. This project was partially funded by the Laboratory Director's Research and Development fund. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan).

Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of funding agencies.

REFERENCES

Chumki Basu, Haym Hirsh, William W. Cohen, and Craig Nevill-Manning. 2001. Technical paper recommendation: A study in combining multiple information sources. *Journal of Artificial Intelligence Research* 14, 1 (2001), 231–252.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research (JMLR)* 3, Jan (2003), 993–1022.

Duen Horng Chau, Aniket Kittur, Jason I. Hong, and Christos Faloutsos. 2011. Apolo: Making sense of large network data by combining rich user interaction and machine learning. In *Proc. the SIGCHI Conference on Human Factors in Computing Systems (CHI)*. 167–176.

Jaegul Choo, Shawn Bohn, and Haesun Park. 2009. Two-stage framework for visualization of clustered high dimensional data. In *Proc. IEEE Symposium on Visual Analytics Science and Technology (VAST)*. 67–74.

Jaegul Choo, Changhyun Lee, Chandan K. Reddy, and Haesun Park. 2013b. UTOPIAN: user-driven topic modeling based on interactive nonnegative matrix factorization. *IEEE Transactions on Visualization and Computer Graphics (TVCG)* 19, 12 (2013), 1992–2001.

Jaegul Choo, Hanseung Lee, Jaeyeon Kihm, and Haesun Park. 2010. iVisClassifier: An interactive visual analytics system for classification based on supervised dimension reduction. In *Proc. the IEEE Conference on Visual Analytics Science and Technology (VAST)*. 27–34.

Jaegul Choo, Hanseung Lee, Zhicheng Liu, John Stasko, and Haesun Park. 2013a. An interactive visual testbed system for dimension reduction and clustering of large-scale high-dimensional data. In *Proc. SPIE 8654, Visualization and Data Analysis (VDA)*. 1–15.

Fan Chung. 2007. The heat kernel as the pagerank of a graph. *Proc. the National Academy of Sciences (PNAS)* 104, 50 (2007), 19735–19740.

Cody Dunne, Ben Shneiderman, Robert Gove, Judith Klavans, and Bonnie Dorr. 2012. Rapid understanding of scientific paper collections: Integrating statistics, text analytics, and visualization. *Journal of the American Society for Information Science and Technology* 63, 12 (2012), 2351–2369.

Khalid El-Arini and Carlos Guestrin. 2011. Beyond keyword search: discovering relevant scientific literature. In *Proc. the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. 439–447.

Khalid El-Arini, Min Xu, Emily B. Fox, and Carlos Guestrin. 2013. Representing documents through their readers. In *Proc. the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. 14–22.

Lars Eldén and Haesun Park. 1999. A Procrustes problem on the Stiefel manifold. *Numer. Math.* 82 (1999), 599–619. Issue 4.

Alex Endert, Patrick Fiaux, and Chris North. 2012. Semantic interaction for visual text analytics. In *Proc. the SIGCHI Conference on Human Factors in Computing Systems (CHI)*. 473–482.

Keinosuke Fukunaga. 1990. *Introduction to statistical pattern recognition, second edition*. Academic Press.

Thomas Hofmann. 1999. Probabilistic latent semantic indexing. In *Proc. the ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. 50–57.

Peg Howland and Haesun Park. 2004. Generalizing discriminant analysis using the generalized singular value decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 26, 8 (2004), 995–1006.

John R. Hurley and Raymond B. Cattell. 1962. The Procrustes program: Producing direct rotation to test a hypothesized factor structure. *Behavioral Science* 7, 2 (1962), 258–262.

Hyunsoo Kim and Haesun Park. 2007. Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis. *Bioinformatics* 23, 12 (2007), 1495–1502.

Jingu Kim and Haesun Park. 2008. Sparse nonnegative matrix factorization for clustering. *Georgia Institute of Technology* (2008).

Jingu Kim and Haesun Park. 2011. Fast nonnegative matrix factorization: An active-set-like method and comparisons. *SIAM Journal on Scientific Computing* 33, 6 (2011), 3261–3281.

Harold W. Kuhn. 1955. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly* 2, 1-2 (1955), 83–97.

Hanseung Lee, Jaeyeon Kihm, Jaegul Choo, John Stasko, and Haesun Park. 2012. iVisClustering: An interactive visual document clustering via topic modeling. *Computer Graphics Forum (CGF)* 31, 3pt3 (2012), 1155–1164.

James R Lewis. 1995. IBM computer usability satisfaction questionnaires: psychometric evaluation and instructions for use. *International Journal of Human-Computer Interaction (IJHCI)* 7, 1 (1995), 57–78.

G. Marchionini and B. Shneiderman. 1988. Finding facts vs. browsing knowledge in hypertext systems. *Computer* 21, 1 (1988), 70–80.

David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. 2010. Automatic evaluation of topic coherence. In *Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics*. 100–108.

Mark EJ Newman. 2004. Fast algorithm for detecting community structure in networks. *Physical review E* 69, 6 (2004), 066133.

Haesun Park, Barry L. Drake, Sangmin Lee, and Cheong H. Park. 2007. Fast linear discriminant analysis using QR decomposition and regularization. *Technical Report GT-CSE-07-21* (2007).

Peter Pirolli. 1997. Computational models of information scent-following in a very large browsable text collection. In *Proc. the SIGCHI Conference on Human Factors in Computing Systems (CHI)*. ACM, 3–10.

Peter Pirolli and Stuart Card. 1999. Information foraging. *Psychological Review* 106, 4 (1999), 643–775.

Catherine Plaisant. 2004. The challenge of information visualization evaluation. In *Proc. the Working Conference on Advanced Visual Interfaces (AVI)*. 109–116.

Catherine Plaisant, Jean-Daniel Fekete, and Georges Grinstein. 2008. Promoting insight-based evaluation of visualizations: From contest to benchmark repository. *IEEE Transactions on Visualization and Computer Graphics (TVCG)* 14, 1 (2008), 120–134.

Purvi Saraiya, Chris North, and Karen Duca. 2005. An insight-based methodology for evaluating bioinformatics visualizations. *IEEE Transactions on Visualization and Computer Graphics (TVCG)* 11, 4 (2005), 443–456.

Noam Slonim and Naftali Tishby. 2000. Document clustering using word clusters via the information bottleneck method. In *Proc. the ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. 208–215.

Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. 2008. ArnetMiner: Extraction

and mining of academic social networks. In *Proc. the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. 990–998.

Chong Wang and David M Blei. 2011. Collaborative topic modeling for recommending scientific articles. In *Proc. the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. 448–456.

James A Wise, James J Thomas, Kelly Pennock, David Lantrip, Marc Pottier, Anne Schur, and Vern Crow. 1995. Visualizing the non-visual: Spatial analysis and interaction with information from text documents. In *Proc. the Information Visualization*. 51–58.

Wei Xu, Xin Liu, and Yihong Gong. 2003. Document clustering based on non-negative matrix factorization. In *Proc. the ACM SIGIR Conference on Research and Development in Informaion Retrieval (SIGIR)*. 267–273.

Xiaohui Yan, Jiafeng Guo, Yanyan Lan, and Xueqi Cheng. 2013. A biterm topic model for short texts. In *Proc. the International Conference on World Wide Web (WWW)*. 1445–1456.