VisIRR: Interactive Visual Information Retrieval and Recommendation for Large-scale Document Data

Jaegul Choo, Changhyun Lee, Edward Clarkson, Zhicheng Liu, Hanseung Lee, Duen Horng (Polo) Chau, Fuxin Li, Ramakrishnan Kannan, Charles D. Stolper, David Inouye, Nishant Mehta, Hua Ouyang, Subhojit Som, Alexander Gray, John Stasko, and Haesun Park

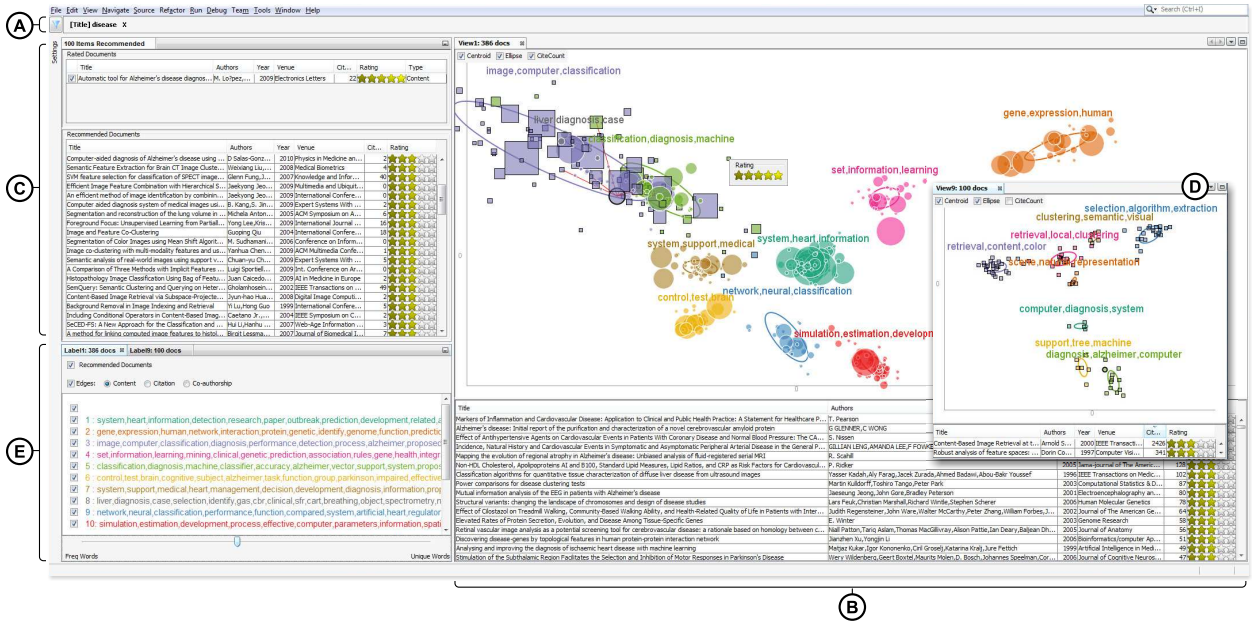


Fig. 1: An overview of the VisIRR system. Given about half a million academic papers in the system, the user can start by issuing a query (A), e.g., a keyword 'disease.' By performing clustering and dimension reduction, VisIRR visualizes the retrieved documents in a scatter plot and a table view (B) along with a topic cluster summary (B)(E). In the scatter plot view, a circular node represents a query-retrieved item, and a rectangular one does a recommended item. Their node size encodes the number of citations. After identifying a few documents of interest, the user can assign them his/her preference in a 5-star rating scale both in a scatter plot and in a table view. Based on this preference feedback, the system now provides a list of recommended items in another table view (C), and furthermore they are projected back to the existing scatter plot view (B) so that the consistent topical perspective can be maintained. To better understand the recommended items, the user can apply 'computational zoom-in' on this set, which gives a clearer scatter plot with a more semantically meaningful summary (D). Finally, the system provides the option to choose different recommendation schemes based on contents, a citation network, and a co-authorship network.

Abstract—We present a visual analytics system called VisIRR, which is an interactive visual information retrieval and recommendation system for document discovery. VisIRR effectively combines both paradigms of passive pull through a query processes for retrieval and active push that recommends the items of potential interest based on the user preferences. Equipped with efficient dynamic query interfaces for a large corpus of document data, VisIRR visualizes the retrieved documents in a scatter plot form with their overall topic clusters. At the same time, based on interactive personalized preference feedback on documents, VisIRR provides recommended documents reaching out to the entire corpus beyond the retrieved sets. Such recommended documents are represented in the same scatter space of the retrieved documents so that users can perform integrated analyses of both retrieved and recommended documents seamlessly. We describe the state-of-the-art computational methods that make these integrated and informative representations as well as real time interaction possible. We illustrate the way the system works by using detailed usage scenarios. In addition, we present a preliminary user study that evaluates the effectiveness of the system.

Index Terms—recommendation, document analysis, dimension reduction, clustering, information retrieval, scatter plot

1 Introduction

- *Jaegul Choo, Fuxin Li, Nishant Mehta, Hua Ouyang, Alexander Gray, John Stasko, Haesun Park. are with Georgia Institute of Technology. E-mail: {joyfull,fli,niche,houyang,agray,stasko,hpark}@cc.gatech.edu.*
- *Changhyun Lee, Duen Horng (Polo) Chau, Ramakrishnan Kannan, Charles D. Stolper, Subhojit Som. are with Georgia Institute of Technology. E-mail: {clee407,polo,rkannan,chadstolper,subhojit@gatech.edu}@gatech.edu.*
- *Edward Clarkson. is with Georgia Tech Research Institute. E-mail: Edward.Clarkson@gtri.gatech.edu.*
- *Zhicheng Liu. is with Stanford University. E-mail: zcliu@cs.stanford.edu.*
- *Hanseung Lee. is with University of Maryland. E-mail: soul3434@gmail.com.*
- *David Inouye. is with University of Texas at Austin. E-mail: davidinouye@gmail.com.*

These days, researchers are faced with a deluge of new papers appearing each day, any of which might potentially contain a new development which could be critical to one of the questions he or she is investigating. The challenge is similar to that of finding an available needle in a haystack each day, with limited attention and time resources.

This problem regime is highly under-explored, compared to the billions that have been invested in the related paradigm of web search. Instead, the researcher or analyst is solving a subtle investigative problem for which each of several documents provides clues. By seeing this as an information retrieval (IR) problem, the focus in this paper is on the long tail, or **recall** (making sure as few relevant documents are missed), while in web search the focus is generally on the quicker gratification of **precision** (making sure the first page of hits or so contain very relevant documents).

In general, search is a form of "**pull**" technology, in which the user takes actions by forming and issuing queries. However, in the former case where a high recall is concerned, what queries to issue, e.g., proper keywords, becomes crucial in order for users to obtain the documents of their interest. As a way to compensate this issue, a **recommendation**, or a "**push**" technology, in which the system finds things of interest to suggest to the individual user, has recently been popular in various domains. Whereas a search engine is more or less stateless and the same for all users, a recommendation system involves personalization, remembering aspects of the state of the user's interests and investigations so far.

In the context of visual analytics, the document analysis has long been one of the main areas studied. Visual analytics systems for document data, such as IN-SPIRE [45] and JIGSAW [41], can help giving an overall understanding about a set of documents as well as revealing their intra-set relationships that would have been difficult and time-consuming without the help of interactive visualization. However, despite the fact that personalized recommendations seem to be a natural fit with interactive visualization in that it directly utilizes the history of user interactions, there are few instances of such work in the visual analytic community.

As one of the milestones to fill this gap, we present a novel document 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 and personalized recommendation. Basically, as seen in Fig. 1, VisIRR adopts a scatter plot as a main visualization form similar to IN-SPIRE. In other words, the documents to be visualized are first clustered into several groups via a clustering algorithm and then projected to a 2D space via a dimension reduction algorithm. However, VisIRR features various novel aspects compared to existing systems, as follows.

- *Efficient large scale data processing*: VisIRR currently handles about half a million documents and scales linearly with respect to newly added documents in terms of the amount of the required computation and memory size.

- *Advanced clustering and dimension reduction techniques*: As core computational modules, VisIRR adopts state-of-the-art techniques such as nonnegative matrix factorization (NMF) for clustering and linear discriminant analysis (LDA) for dimension reduction. These techniques give the results with a much better quality as well as with faster computational time than traditional methods including k-means, principal component analysis (PCA), multidimensional scaling. Additionally, VisIRR provides an alignment capability for both clustering and dimension reduction to facilitate easy comparisons between different visualization snapshots.

Manuscript received 31 March 2013; accepted 1 August 2013; posted online 13 October 2013; mailed on 27 September 2013.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

- *Preference-based personalized recommendation*: In addition to exploratory analysis of query-retrieved results, VisIRR supports recommendation of potentially interesting documents to users based on the preferences users assign to documents. This recommendation enables users to discover those documents users' query processes cannot reveal easily. The back-end recommendation module, which is based on PageRank-style graph diffusion algorithm [33], performs efficiently with large-scale data.

To integrate all these capabilities into a mature visual analytics system, we incorporate various building blocks for front-end GUI's and back-end computational algorithms. This paper mainly presents these building blocks in more detail with detailed usage scenarios. The rest of this paper is organized as follows. Section 2 discusses related work. Section 3 explains the front-end GUI modules and comprehensive usage scenarios that highlight the key capabilities of the system. Afterwards, Section 4 mainly discusses how we efficiently handle all the necessary information from a large-scale data corpus with a scalable expansion, and Section 5 describes computational methods used in the back-end of the system. Section 6 briefly presents the user study we conducted to evaluate the system. Finally, Section 7 concludes the paper and discusses about the future work.

2 Related Work

Information seeking behavior is a complex human activity, and one that varies dramatically with system capabilities and user's model of those capabilities [30]. Ill-defined document search tasks such as literature searches are often termed 'exploratory search' tasks, in contrast with more defined tasks such as finding a known, specific item from among a set. In the past, traditional information retrieval has focused much more on the latter than the former.

In the context of exploratory interfaces, information foraging [35] and scent theory [34] suggest making clusters of related data clear and facilitating the process of finding new clusters of interest. To that end, many search result visualization systems also work in concert with automated clustering algorithms, especially when the information space is extremely large or unstructured. The Pacific Northwest Lab's SPIRE system (and IN-SPIRE follow-on) uses clustering to extract common themes, and includes several visualization components [45]. Its Themescape component is an abstract 3D landscape depiction of a document space, with arrangements of hills and valleys representing the relatively strength of various themes in the document corpus and how those themes interrelate. Other systems have used this general clusters-in-landscapes (both 2D and 3D) as well [39, 7, 4]. iVisClustering [27] is an interactive document clustering system focused on the user interactions to improve cluster quality based on an advanced technique called latent Dirichlet allocation [6]. On the other hand, rather than providing user interactions customized to a particular clustering technique, the Testbed system [14] offers a wide variety of clustering algorithms and easy comparisons between them via an alignment process VisIRR has adopted.

Using visualization for exploring text data is an active research area within and among many fields. Here, we highlight only a sample of relevant work from different areas and refer the reader to a recent survey of visual text analytics [3] for a more comprehensive treatment.

Unsurprisingly, visualization of document collections as been explored for some time in library science. A relatively early example is the Envision digital library, which includes a visualization system that places documents in a 2D grid according to user-selectable attributes [32]. Systems have used various information visualization techniques such as hyperbolic trees [22, 40] and treemaps [19, 16] to visualize results. Curated collections such as those found in digital libraries more often have pre-formed hierarchies to leverage in visual analytics applications, but simple clustering methods have been implemented as well [40].

When document categories and groupings are not already extant, automated methods of clustering and classifying collections are key to exploratory tools, including those supporting visual analysis. A recent survey [3] distinguishes between the visualization of a single

document (e.g., tag clouds) vs. a document collection and between time- (e.g., TIARA [29]) and network-oriented collection systems. Because VisIRR's clustering system implicitly creates relationships among members (and its graph diffusion-based recommendation system explicitly uses such data), examples of the last category are most relevant. Jigsaw [42] visualizes network relationships between documents and various entities, e.g., actors, events, etc., automatically extracted from them.

A recently proposed Apolo system [10] uses a mixed-initiative approach that bootstraps initial user-specified categories and classifications into more comprehensive system-suggested categorization of new documents. However, Apolo is exemplar-based method where the user is assumed to clearly have a few of documents of their interest. In this sense, Apolo mainly supports a bottom-up style of analyses. On the contrary, VisIRR initially takes a top-down approach in that it initially starts from an overview visualization of a potentially fairly large amount of documents retrieved by user queries. Once the documents of the user's interest is identified, however, VisIRR supports also a bottom-up style approach via recommendation processes based on the user preferences on particular documents, thereby gradually expanding the user's scope beyond the query-retrieved set.

There has been significant commercial and academic interest in the topic of exploratory search for scientific literature itself for some time. Several commercial tools are targeted to this problem, with a variety of automated and visual features. Google Scholar [1] automatically extracts research works and their citation networks, but has few visual or recommendation features. The Microsoft Academic Search system from Microsoft Research [2] is a similar offering that also includes more advanced network-style visualization of authorship connections as well various ways of examining topical, institutional and venue trends and rankings.

Direct introspection of the academic research process has been a common topic in academia as well. One variation is automated recommender/matching systems, often applied to the problem of matching individual papers from a corpus to individuals from a slate of candidate reviewers [5, 44]. More relevant to VisIRR are those systems that are more exploratory or analytical in nature. The Action Science Explorer (ASE) [17] focuses on co-citation network visualization, with document clusters created manually or by heuristic [31]. It also includes full-text citation context features not available to VisIRR. The FacetAtlas system [9] automatically clusters document collections using a Kernel density estimation algorithm and provides multi-faceted links between document nodes (rather than just keyword or author searches as in VisIRR). CiteSpace II [11] is a visual tool for identifying new or old research trends in a given set of documents (assumed to be a relatively coherent set produced by a keyword query on a large corpus).

However, none of these systems include one of VisIRR's key contributions: *a user-driven recommendation system that explicitly includes relevant documents from the larger search space vs. a dramatically reduced one from an initial search query*.

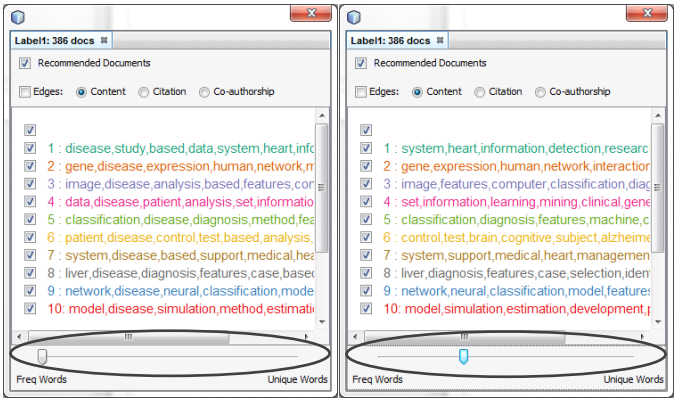
3 VisIRR Design and Function[1]

In this section, we briefly introduce the user interfaces of VisIRR and describe example analysis scenarios to demonstrate how VisIRR works in detail.

3.1 User Interface

The user interface of VisIRR is mainly four parts. The *Query Bar* at the top (Fig. 1(A)) enables users to issue queries dynamically using various fields such as a keyword, an author name, a publication year, and a citation count. The *Scatter Plot view* (with document details shown in the lower table) (Fig. 1(B)) visualizes the retrieved documents (as well as any recommended documents) with their cluster summary labels. The color and the size of each node in a scatter plot represent the cluster it belongs and its citation count, respectively.

[1]A high-quality video introducing VisIRR is available at `http://www.cc.gatech.edu/~joyfull/vast13/visirr/visirr_final.html`



(a) Default cluster summary (b) Distinct cluster summary

Fig. 2: A Comparison between default and distinct cluster summaries. Since all the documents include the query word "disease", most clusters contain this word as one of the most frequent keywords (a). By adjusting the slider of *common-vs-unique words* in the *Label panel*, the cluster summary shows much clearer meanings (b).

Such a 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 the documents whose ratings have been assigned by users (Fig. 1(C) upper table) as well as the resulting recommended documents (Fig. 1(C) lower table). These recommended documents are also visualized in the *Scatter Plot view* as rectangles while the query-retrieved documents are shown as circles. Finally, the *Label panel* provides additional controls such as highlighting and/or hiding particular clusters, changing how cluster summary labels are chosen, and showing direct edge relationships from rated documents to their system-derived recommended documents (Fig. 1(E)).

3.2 Usage Scenarios

VisIRR has been implemented using a modified version of the ArnetMiner dataset, 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 detail in Section 4. The following scenarios illustrate the utility of VisIRR for tasks related to this dataset.

3.2.1 A Visual Overview of Query-Retrieved Documents

The user starts by issuing queries from the *Query Toolbar*. Suppose the user issues a query of keyword "disease" from a title field. Once documents are retrieved due to this query, the clustering and dimension reduction steps are performed to generate the *Scatter plot view* (Fig. 1(A)). Since most clusters contain the keyword "disease", the user can adjust a slider in the *Label panel* in order to obtain more distinctive cluster summaries, as shown in Fig. 2. From the *Scatter plot view*, the user can drill down to a cluster of interest, e.g., the clusters about gene expression data (the top right), image analysis (the top left). By moving a mouse pointer to a data point, the user can check the document details via a tooltip text and also skim through the document list in the lower table, which is by default sorted by the number of citations. The user can also pan and zoom to enlarge a particular cluster or area of interest.

3.2.2 Drilling Down via Computational Zoom-in

Now, the user can drill down a particular cluster via an interaction we call *computational zoom-in*. The computational zoom-in enables the user to select an arbitrary subset of documents by visualizing them as a separate view with their own clustering and dimension reduction results. These subsets can be, for example, particular clusters when their semantic meanings are not clear involving multiple topics. On the

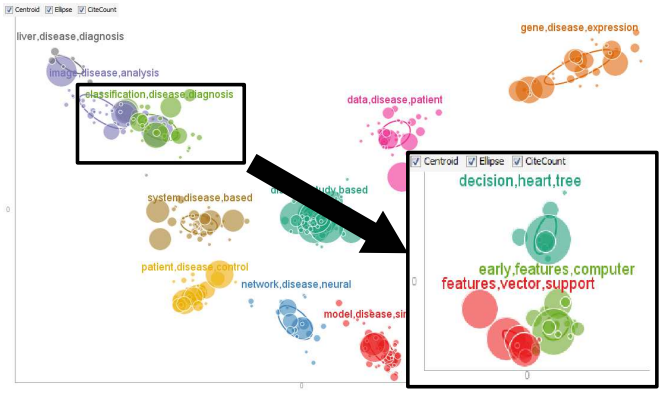
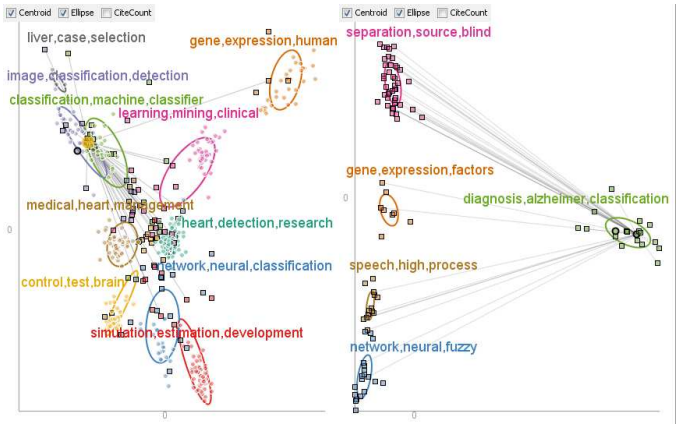


Fig. 3: An example of the computational zoom-in interaction. For a user-selected region (black rectangle on the top left), this interaction provides a separate view by involving only these points to compute their own cluster summary and dimension reduction coordinates. The resulting view now shows a clear overview about these cluttered data, revealing detailed clusters about 'support vector machines' and 'decision trees' typically applied in medical image analyses (black rectangle on the bottom right).



(a) A visualization of retrieved and recommended documents

(b) A visualization of only the 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.

other hand, the user can select a cluttered region where many points are mixed together.

Fig. 3 shows an example of the computational zoom-in interaction. After performing computational zoom-in on a highly cluttered area in an original view (black rectangle on the top left), the resulting view successfully reveals several clear clusters e.g., the one about 'support vector machines' and another about 'decision trees' typically applied in medical image analyses (black rectangle on the bottom right).

3.2.3 Dynamic Queries and Multi-view Alignment

In addition to exploring visualized clusters, the user can apply additional queries to further narrow down the retrieved document set. Suppose the user wanted to focus on those recently published in 2008 or later and thus created another filter from the *Query Toolbar* in conjunction with the previous keyword query "disease." Given such a new set of documents, VisIRR creates another visualization with its own clustering and dimension reduction. The user could then compare between the new and the 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 were more/less popular from 2008. However, since the cluster colors and the dimension reduction results have been computed independently, it is not straightforward to easily compare these differences based on the visualization results.

To solve this problem, once a new visualization is created, VisIRR performs an alignment step on the new clustering and dimension reduction results with respect to the previous visualization result so that the visual coherences in terms of the cluster colors and the spatial coordinates of data points can be maintained. The algorithm details are discussed in Section 5.3. For instance, as opposed to an unaligned visualization in Fig 4(a), an aligned one in Fig 4(c) is shown to be much easier to compare against the previous visualization shown in (Fig 4(b). From the aligned visualization, the user can easily see that the cluster about *outbreak detection*, shown as a green cluster in the middle of Figs. 4(b)(c), was not actively studied from 208.

3.2.4 Content-based Recommendation

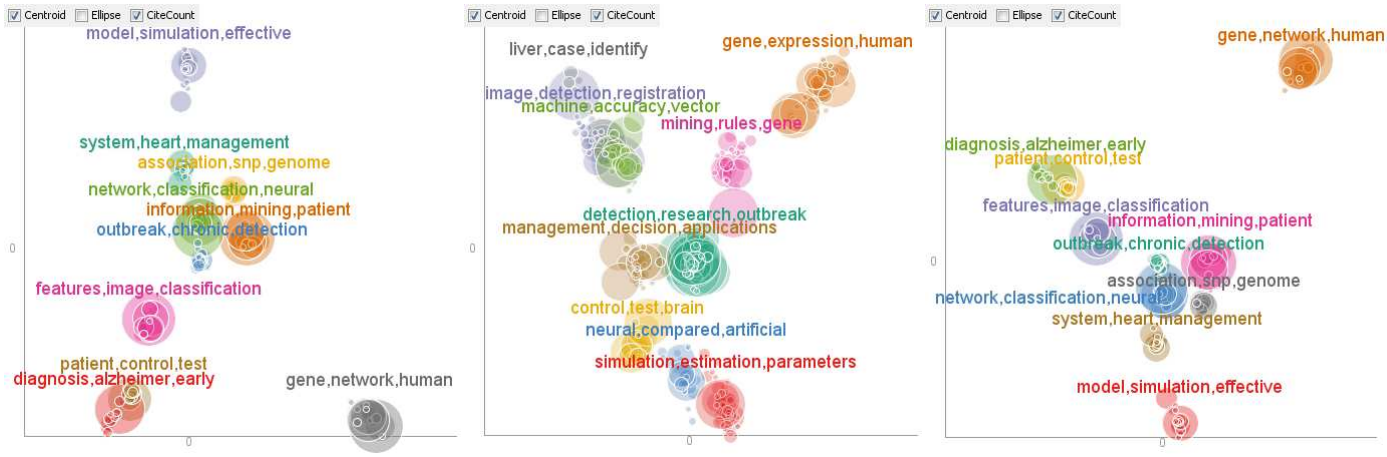
Throughout analyses, the user can assign ratings to the documents he/she likes or dislikes. Among the retrieved documents, suppose the user found a document "Automatic tool for Alzheimer's disease diagnosis using PCA and Bayesian classification rules" interesting and assigned a 5-star rating(highly-like) by right-clicking the corresponding data point in the *Scatter Plot View*. Based on this user preference information, VisIRR identifies the recommended documents based on the content similarity. These rated and the recommended documents are displayed in a tabular form in the *Recommendation view* (Fig. 1(C)).

From the list of recommended documents shown in the lower table, the user could obtain an idea that the research about Alzheimer's disease mainly involves an image analysis, clustering, classification, etc. Notice that without such a recommendation capability of VisIRR, the user would not be able to obtain these documents since these documents was not included in the retrieved set by user queries. In the *Scatter Plot view*, the user can see these recommended documents at the upper left corner around the rated document and its nearby clusters. To obtain a better idea about the recommended documents, the user can create another visualization only using this subset with a new clustering and dimension reduction (Fig. 1(D)). From its own cluster summary and visualization, the user could see that the documents directly related to Alzheimer's disease are mainly shown in the bottom half while the upper half in the *Scatter Plot view*, shows those mainly related to image analysis such as content-based image retrieval, clustering, etc.

3.2.5 Citation- and Co-authorship-based Recommendation

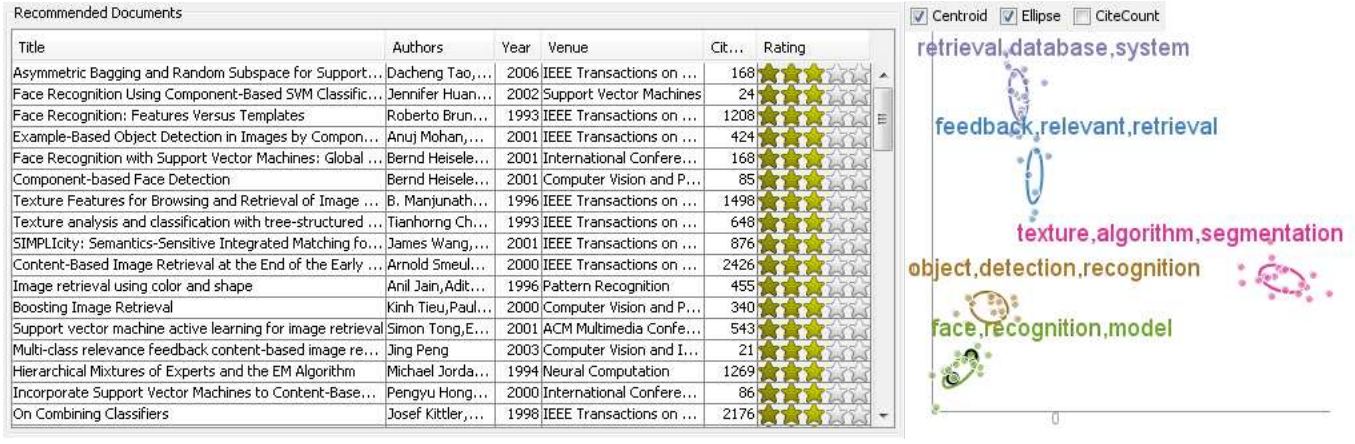
Now, among the recommended documents, the user chose another document "Automatic Classification System for the Diagnosis of Alzheimer Disease Using Component-Based SVM Aggregations" and assigned it a 5-star rating. This time, the user changes its recommendation type to a citation-based one from the *Recommendation panel* in order to obtain highly-cited documents relevant to this document. As expected, VisIRR's top-ranked recommended documents are relatively highly cited papers, as shown in Fig. 5(a). After generating another visualization only using these recommended items, the user can obtain a summary about them, the clusters of which are composed of image retrieval, object detection/recognition, face recognition, and texture analyses (Fig. 5(b)). Notice that these types of recommendation results would not be easily obtained by a simple keyword search since these recommended documents do not contain specific keywords in common. Instead, they are only implicitly related with each other via a citation network on which VisIRR can perform a recommendation based.

In addition, the user also wanted to know what other topics or areas the authors of this paper are involved in. To this end, the user changed the recommendation type to a co-authorship-based one from the *Recommendation view*. In addition, to better show the direct co-authorship relationships from the rated paper, the user turned on the



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

Fig. 4: Effects of clustering and dimension reduction alignments. A reference view (b) shows the documents with a query word "disease" while the other two views (a)(c) contain the subset of them published from year 2008 with their own clustering and dimension reduction steps applied. For an unaligned view (a), it is difficult to compare against the reference view since there is no correspondence in terms of the coordinates of data points and clusters. However, in an aligned view (c), the clusters match those in the reference, and their spatial correspondences in the scatter plot are maintained.



(a) The top-ranked recommended document list (b) A visualization 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." VisIRR recommends various papers mostly with high citation counts, which are relevant to the rated paper.

"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 the direct co-authorship relations from the rated document, as shown in Fig. 6(a). Similar to the previous case, the user can generate another visualization of only the recommended items to have a better idea about the recommended documents. After varying the number of clusters, the user obtains a new visualization as shown in Fig. 6(b). From this visualization, the user could gain an insight that the authors of the rated paper have written the papers, other than Alzheimer's disease-related papers (the green cluster on the right), in the four areas corresponding to blind source separation, gene expression, speech processing, and neural networks. This potentially indicates that the user, who was originally interested in Alzheimer's disease diagnosis, could expand his/her research by following the way the authors of the rated paper have published in other domains.

4 Data Collection / Ingestion

4.1 Initial Data Collection

VisIRR is intended to efficiently handle a large-scale document corpus with a rich set of features. To this end, VisIRR begins with the ArnetMiner data set, which is composed of about half a million academic papers, books, etc. [43].[2] Although the data set is mainly used in citation network analyses, it includes a variety of both structured and unstructured information such as a title, keywords, an abstract, authors, a publication year, a venue, a document type such as a book, a paper, etc., papers in the reference list, papers citing this document, the number of references, the number of citations.

However, the original data set has numerous missing values and inconsistencies such as different expressions of an author name, a publication venue, etc. To clean up the data, we utilize the Microsoft Academic Search API's.[3] Specifically, we used a title of each document as a query in order to obtain the full information about the document from the Microsoft Academic Search API, which fills the missing values and rectifies inconsistencies. Finally, VisIRR builds upon 432,605 documents spanning from year 1825 to 2011.

4.2 Data Ingestion

Now we describe how we make these large-scale data readily available for real-time interactive analyses in VisIRR. Basically, VisIRR maintains the information about data in three different forms, (1) original fields of data, (2) a vector representation, and (3) a graph representations, in an efficient and scalable way. In order to efficiently manage the large-scale data in all these various forms, we carefully optimized various data processing/storage techniques via database construction, pre-computation of frequently used information, balanced storage between disk and memory. Eventually, the system is easily and widely deployable in typical commodity PC's instead of requiring high-performance parallel machines.

4.2.1 Original Field of Data

For efficient and flexible query support, we have encoded the original data as a SQL database including full-text search capabilities on a title, keywords, an abstract, and a venue fields. For clustering and dimension reduction steps, we have pre-computed the sparse vector representations of individual documents based on a title, keywords, and an abstract fields together via a bag-of-words encoding scheme. Each vector representation is stored as a single file in a disk, the file name of which is the document ID. In this way, VisIRR can retrieve the vector representations of documents using their document ID's in the time complexity of $O(1)$.

[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`.

4.2.2 Vector Representation

Once the vector representations of documents are loaded into a memory, VisIRR manage them in a similar way to cache replacement algorithms. That is, the vector representations already loaded into the memory is referenced from the memory whenever needed. When the total memory-loaded vectors exceed a pre-defined maximum memory size, the least recently used vectors are removed from the memory. When needed later, they are loaded from a disk once again. This way, VisIRR does not need to load the vector representations of all the documents from the beginning, which will take significant time and memory at the system startup. At the same time, VisIRR prevents the required memory size from blowing up due to a long-term usage of the system.

4.2.3 Graph Representation

The recommendation module, which will be described in Section 5, requires an input graph where the nodes correspond to documents and the edges represent their pairwise similarities/relationships. We have pre-computed three such graphs for the entire data set using contents, a citation network, and co-authorship, respectively, in order to support various recommendation capabilities. For content-based graph, we initially computed the pairwise cosine similarities between all the pairs of documents using their vector representations. Since maintaining all the pairwise information requires $O(n^2)$ storage where n is the total number of documents, we identified the fixed number (10 in our case) of the most similar documents for each document and kept only the edges between them. For citation graph, we formed edges between a pair of documents if either cites the other. For co-authorship graph, edges are created if two documents share the common author(s). Since citation and co-authorship graphs are typically sparse, we stored all these edge information. For each graph, VisIRR maintains the mappings from an individual document to a 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 the time complexity of $O(1)$.

4.3 Scalable Update for New Data

Even though VisIRR already contains a large-scale data of about half a million documents, it is crucial to have a capability to efficiently update the above-described information including newly added documents. An updating process is composed of two parts: updating the information about existing documents and obtaining the representations of new documents. First, in the case of the original fields of data, the information about new documents can be easily added to the database without affecting the existing data. Second, In the case of updating bag-of-words vector representations, new documents generally causes newly appearing keywords to be indexed as additional dimensions. However, sparse vector representations of existing documents would still remain the same, and thus we only need to compute the representation of new documents, which can also be easily done.

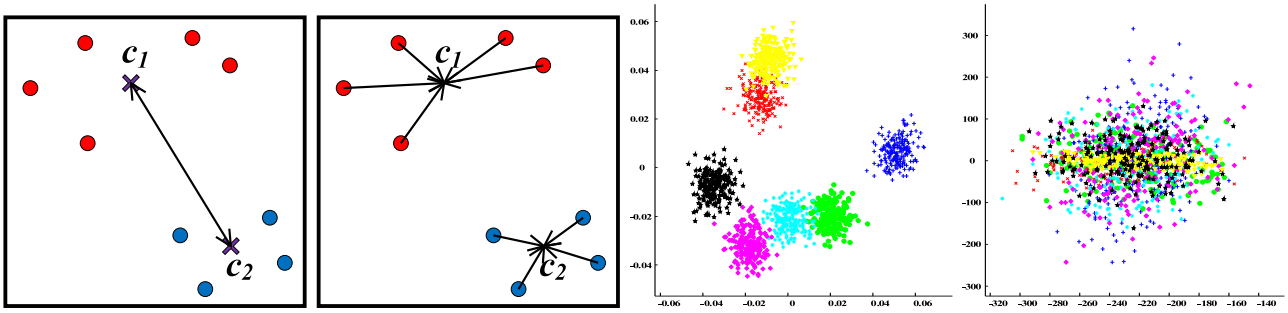
Finally, in the case of updating graph representations, the only tricky part is to update the content similarity graph, where the top 10 most similar documents and their cosine similarity values are maintained. Specifically, we have to compute the pairwise similarity between all the existing documents and all the new documents, and then compare these similarity values against the current top 10 similarity values. If any of the former ones are greater than the latter ones, the corresponding edges are replaced with those to the new documents. The computational complexity of this process is $O(n \times n_{new})$ where n and n_{new} are the numbers of the existing and the new documents, respectively.

5 Computational Methods

The key computational methods in VisIRR are clustering, dimension reduction, alignment, and graph-based recommendation. In this section, we describe each module in detail.

5.1 Clustering

Clustering plays a crucial role in providing a summary of a given set of documents as a manageable number of groups based on their semantic



(a) Maximization of distances between cluster centroids (b) Minimization of approximate cluster radii (c) LDA (d) PCA

Fig. 7: A high-level idea of LDA and a comparison example betwee LDA and PCA. A different color corresponds to a different cluster, and c_1 and c_2 are the cluster centroids. LDA tries to find a reduced-dimensional representation of data by putting different clusters as far as possible (a) and representing each cluster as compact as possible (b). (c) and (d) show an example 2D scatter plots obtained by PCA and LDA, respectively, for artificial Gaussian mixture data with 7 clusters and 1,000 original dimensions. From a comparison between them, LDA is shown to reveal a much clearer cluster structure than PCA in a 2D space.

meanings. The resulting cluster indices are used to color-code documents in a scatter plot with their cluster summaries in terms of the most frequently shown keywords (Fig. 1(B)(E)). VisIRR adopts a state-of-the-art technique called nonnegative matrix factorization (NMF) [23], which have shown superior performances in document clustering over traditional methods such as *k*-means [24, 46].

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. NMF can be formulated using the Frobenius norm as

$$\min_{W,H \geq 0} \|X - WH\|_F^2 . \tag{2}$$

In the context of document clustering, each column vector $x_i \in \mathbb{R}^{m \times 1}$ of X represents each document as an m-dimensional vector via a bag-of-words encoding, with some additional pre-processing steps such as inverse-document frequency weighting and vector norm normalization. The value of k represents the number of clusters. For clustering, one can utilize H as a soft clustering vector representation of documents. That is, the column vector $h_i \in \mathbb{R}^{k \times 1}$ of H represents such a soft clustering vector for the i-th document, and by taking the index the value of which is the largest, the cluster index of the document can be obtained.

The specific NMF algorithm we have used is based on a recently proposed block principal pivoting algorithm [25],[4] which is found to be one of the fastest and reliable algorithms. Although not reported, we have conducted an extensive amount of comparison of NMF against traditional clustering techniques such as *k*-means, and we found that NMF mostly gives semantically more meaningful clusters than any other methods while requiring a significantly faster computational time.

5.2 Dimension Reduction

Given high-dimensional vector representations of documents, dimension reduction computes their 2D representations so that they can be visualized in a scatter plot (Fig. 1(B)). From the scatter plot, users can get an idea about how clusters/documents are related with each other. VisIRR adopts an advanced dimension reduction method called linear discriminant analysis (LDA) [20].

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

Unlike traditional methods such as principal component analysis and multidimensional scaling, LDA explicitly utilizes additional cluster label information, which are taken from the clustering module, associated with the input high-dimensional vectors. Using this information, LDA tries to preserve the cluster structure in the low-dimensional space by such that the dimension-reduced result can clearly reveal the underlying cluster structure in the input data. In this manner, as shown in Fig. 7, LDA has an advantage over most traditional methods such as PCA and MDS in that it can provide a clear cluster structure in the data when the cluster label information is given.

Furthermore, VisIRR provides a slider interface for controlling how compactly each cluster is represented by using regularization on LDA, which enables users to focus their analyses at either a cluster level or an individual document level. For more details, refer to [12, 13].

5.3 Alignment

In VisIRR, users can create multiple scatter plots for (1) new parameter values, e.g., the number of clusters in NMF, a regularization value in LDA, and (2) a new set of data from different queries or arbitrary selection by users. In order to maintain consistency between different scatter plots and facilitate their easy comparisons, VisIRR provides alignment capabilities on different clustering and dimension reduction results. By aligning clustering results, users can expect that the same cluster index and color indicates semantically similar meanings. On the other hand, by aligning dimension reduction results, users can expect that the same data point is located in a similar position in the 2D space between different scatter plots.

To align different clustering results, VisIRR utilizes the Hungarian algorithm [26]. Given two sets of cluster assignments for the same set of documents, the Hungarian algorithm finds the optimal pairwise matching of cluster indices between the two sets so that the number of common data items within matching cluster pairs can be maximized. Based on the resulting matching, VisIRR changes the cluster indices and the colors of the newly created scatter plot with respect to those of the used reference scatter plot. In this manner, VisIRR maintains the cluster indices/colors with their consistent semantic meanings throughout multiple visualization results.

The alignment of different dimension reduction results is based on Procrustes analysis [21, 18], which best maps one results to the other with only a rotation matrix. In addition, VisIRR extends the original Procrustes analysis by incorporating translation and isotropic scaling factors as well. That is, given two reduced-dimensional matrices X, $Y \in \mathbb{R}^{m \times n}$, where m is the number of dimensions and n is the num-

ber of data points, VisIRR solves

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

where $Q \in \mathbb{R}^{m\times m}$ is an orthogonal matrix (for rotation), μ_X and μ_Y are m-dimensional column vectors (for translation), k is a scalar (for isotropic scaling), and 1_n is an n-dimensional column vector whose elements are all 1's. Eq. (3) is efficiently solved by using eigendecomposition. These alignment functionalities help users understand how similarly/differently the corresponding data items/clusters are placed between different views.

5.4 Recommendation

The main input to the recommendation algorithm is personalized preferences to particular documents, which are interactively assigned by users in a 5-star rating scale, as shown in the bottom-right in Fig. 1(B). By default, all the documents are assumed to have a 3-star rating, which is converted to a zero preference value, but users can interactively assign ratings to particular documents, where a 1-star corresponds to a preference value of -2, and 5-star to +2, etc.

Given these user preference information, VisIRR identifies the recommended documents by performing a PageRank-style graph diffusion algorithm on a weighted graph of the entire document set. As briefly discussed in Section 4, such a graph can be based on either contents, a citation network, or co-authorship depending on users' choice. Particularly, VisIRR has adopted a heat-kernel-based algorithm [15], which gives a much faster convergence than the other 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 such that its sum is equal to one, and a user preference vector $p \in \mathbb{R}^{N\times 1}$, where the i-th component p_i is the preference value, VisIRR computes the recommendation score vector $r \in \mathbb{R}^{N\times 1}$ of N documents

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

where α and n are user-specified parameters, e.g., by default, $\alpha = 0.7$ and $n = 3$. An intuitive explanation of this formulation is that the preference value $p_i$of node i is propagated to its neighbor nodes with the corresponding weights specified in the graph W at the first iteration, and then the resulting values are then propagated again with the same graph W with the scale factor $(1-\alpha)$ at the next iteration, and so on. Finally, those values computed from each iteration is added up, forming a final recommendation score vector r. Once the computation is done, VisIRR presents the documents with the biggest scores in r as the recommended ones.

One may think that Eq. (4) is computationally intensive because our input graph W is very large-scale. However, all the computations, which are basically matrix-vector multiplications, are performed based on sparse representations. Therefore, as long as W and p have few non-zero entries, the computation is typically done fast. Furthermore, 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 their individual interactions, which essentially makes p have only one non-zero entry. In this way, VisIRR maintains the real-time efficiency of computations during users' frequent interactions.

5.5 Implementation

The system is mainly implemented in JAVA for front-end UI and rendering modules, which are partly based on the FODAVA testbed system [14]. NetBeans Rich Client Platform and IDE[5] have been used for flexible window management. The back-end computational modules NMF and LDA are originally written in MATLAB but we have wrapped them into a JAVA library by using a Matlab built-in functionality called 'Javabuilder.'[6] Since the library made in this manner is self-contained, VisIRR does not require an actual Matlab to be installed. For querying and accessing with the database, we have used H2 library.[7]

[5] http://netbeans.org/features/platform/index.html
[6] http://www.mathworks.com/products/javabuilder/

6 Confirmatory User Study

The evaluation of information visualization and visual analytic systems has been an acknowledged challenge for some time [36]. Insight-based evaluation [38, 37] has gained popularity recently as an alternative to traditional time-and-accuracy measures. As a preliminary gauge of how well our usage scenarios match real user behaviors, we conducted a evaluation of VisIRR with end users, which consisted of an informal, non-experimental insight-based protocol.

The design of this study is evidence-by-existence; that is, provide some support of our implicit VisIRR design claims. For example, show that recommendations outside the initial query set are useful to some people and they can find useful documents with VisIRR. It is not an experimental design as it includes no control condition, so we cannot and do not make any relative claims about VisIRR's effectiveness compared to other research or commercial alternatives (e.g., Google Scholar). Instead, its purpose is more modest: demonstrate VisIRR *can* meet its intended purpose for real users (providing evidence that our imagined user scenarios above are valid), and provide direction for a future, comprehensive experimental or quasi-experimental design.

6.1 Method and Limitations

Participants in the study used VisIRR implemented with the same ArnetMiner-based set of academic articles described in the usage scenarios above. After completing a consent form and a brief demographics questionnaire, they were provided a live demo of the system usage scenario (lasting 5-10 minutes, depending on questions). Participants then used the system to conduct searches of their own choosing and to complete a set of pre-defined tasks concerning 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) [28] along with a few other subjective assessment questions specific to VisIRR.

The system was installed on a workstation with two 2.5GHz Intel Xeon processors and 128GB running 64-bit Windows 7, though the Java VM memory limit was set to only 8 GB. It was connected to both a 30" monitor (1920x1200) and a 19" monitor (1280x1024); users were free to arrange windows on either monitor, but most chose to use the majority of the 30" screen for the VisIRR windows and dialogs with the task response window on the 19" screen.

We recruited 7 male Ph.D. students between the ages of 24-40 enrolled in various technical degree programs (engineering, computer science, robotics). As such, they all had experience doing academic literature searches using online resources such as Googles, Google Scholar, the IEEE/ACM digital libraries, etc. We asked participants to self-rate their familiarity with information visualization and ubiquitous computing literature; all self-rated 4 or less on a 7-point Likert scale for information visualization and 6 of the 7 did so for ubiquitous computing. Participants completed tasks for the area with they were less familiar. The VisIRR system was instrumented to log the UI actions shown in Table 1. We non-intrusively observed users while they completed the tasks.

We present only a few quantitative measures in our results and no mean values as the limited sample and non-experimental nature of the study would render them specious. The tooltip counts in Table 1 are somewhat exaggerated because the VisIRR tooltips have a very short timeout triggering their appearance, meaning many tooltips could be triggered just from panning over one of the document lists or through the scatterplot.

6.2 Results and Discussion

Table 1 shows the raw action counts across all 7 users and all tasks. Those counts match our subjective impressions of watching users

[7] http://www.h2database.com/html/main.html

Table 1: The study UI action counts across all participants and tasks.

Action	Description	Count
Tooltip	A tooltip showing document details triggered by hovering over a table row or scatterplot node.	38897
Rating	The user picks a non-default 1-5 star rating from table entires or scatterplot nodes	80
Details	The user shows the details dialog box for one or more articles	146
Copy	The user copies document information to the clipboard	35
Filter	The user performs a filter (by keyword, year, citation count or author name) on the current results	24

complete tasks: they consistently made use of the major VisIRR features (visualization, ratings and recommendations and details-on-demand). Since one of our most basic questions was whether users would actually make use of the more novel features like ratings and recommendations, this preliminary result is encouraging.

All users made at least 9 distinct document ratings (again, across all tasks), and interestingly did so relatively evenly from different portions of the UI (the recommended, rating and query lists, and the scatterplot). Document details were disproportionately triggered from the visualization (112/146), indicating both that participants interacted with the visualization and drilled down into document details from there. This matches both our subjective observations and post-test user comments like "It's good to have that first clustering result ... It's easy to go deeper down from one or two clusters." Unfortunately, the logging does not distinguish between regular and recommended document nodes in the scatter plot.

On the subjective CSUQ, scores were generally 5 or higher, with the lowest rated scores coming on the questions "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 recover easily and quickly". We suspect these ratings reflect occasional software bugs and crashes that occurred during some participant sessions.

Our results also suggest a potential interesting contrast in user behavior with more traditional keyword search algorithms: one might expect in exploratory tasks with keyword engines to see multiple iterations of keyword refinement and result inspection for a given task or user. However, our users performed relatively few filter actions (all keyword refinements rather than by author, time or citation). However, because VisIRR recommendations expand the search query outside its original bounds (and highlight those nodes which are outside those bounds), iterating keyword terms is less necessary, though future work is necessary to confirm this idea, or to gauge whether this approach is more or less effective than keyword refinement.

Of course, we would hypothesize that rating-based refinement is more productive since it does require less user expertise at generating useful keyword sequences; at least one user agreed, saying that VisIRR "... is definitely much better than blindly search Google Scholar or basic search engines using just a few keywords."

7 Conclusion and Future Work

In this paper, we have presented VisIRR, a visual analytics system called VisIRR, an interactive visual information retrieval and recommendation system for document discovery. One of the primary contributions of VisIRR is that it has effectively combined both paradigms of passive query process and active recommendation by reflecting the user preference feedback. In addition, VisIRR directly tackles a large-scale document corpus via efficient data management and new data updating as well as a suite of state-of-the-art computational methods such as NMF, LDA, and graph diffusion-based recommendation.

Our future work includes the following.

- *Collaborative filtering-based recommendation*: In addition to the preference-based recommendation we have taken, it would be more effective if VisIRR could support collaborative filtering-based approach [8] by using multiple other users' preference information. However, collecting these preference information from various users is sometimes not easy. In this respect, VisIRR could conversely be used as an easy visual interactive tool to collect these preference information after deployed to many users, just as we have collected various information about the user interaction history in Section (6).

- *Fast interactive clustering and layout*: We found that many users often complained about visualization not coming up immediately due to high computation time. When hundreds or thousands of documents are involved, the clustering and the dimension reduction computation typically takes from a few seconds to a minute. In addition, the user sometimes wanted to move documents/clusters see what other documents/clusters move correspondingly. The fast and interactive clustering and layout algorithms incorporating these user feedback would help VisIRR substantially.

8 Acknowledgements

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

References

[1] Google scholar. `http://scholar.google.com`. Accessed: Mar. 2013.

[2] Microsoft academic search. `http://academic.research.microsoft.com/`. Accessed: Mar. 2013.

[3] A. B. Alencar, M. C. F. de Oliveira, and F. V. Paulovich. Seeing beyond reading: a survey on visual text analytics. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(6):476–492, 2012.

[4] K. Andrews, C. Gutl, J. Moser, V. Sabol, and W. Lackner. Search result visualisation with xfind. In *User Interfaces to Data Intensive Systems, 2001. UIDIS 2001. Proceedings. Second International Workshop on*, pages 50–58. IEEE, 2001.

[5] C. Basu, W. Cohen, H. Hirsh, and C. Nevill-Manning. Technical paper recommendation: A study in combining multiple information sources. *arXiv preprint arXiv:1106.0248*, 2011.

[6] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, March 2003.

[7] K. Borner, A. Dillon, and M. Dolinsky. Lvis-digital library visualizer. In *Information Visualization, 2000. Proceedings. IEEE International Conference on*, pages 77–81. IEEE, 2000.

[8] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, UAI'98, pages 43–52, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[9] N. Cao, J. Sun, Y.-R. Lin, D. Gotz, S. Liu, and H. Qu. Facetatlas: Multifaceted visualization for rich text corpora. *Visualization and Computer Graphics, IEEE Transactions on*, 16(6):1172–1181, 2010.

[10] D. H. Chau, A. Kittur, J. I. Hong, and C. Faloutsos. Apolo: making sense of large network data by combining rich user interaction and machine learning. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, pages 167–176. ACM, 2011.

[11] C. Chen. Citespace ii: Detecting and visualizing emerging trends and transient patterns in scientific literature. *Journal of the American Society for Information Science and Technology*, 57(3):359–377, 2006.

[12] J. Choo, S. Bohn, and H. Park. Two-stage framework for visualization of clustered high dimensional data. In *IEEE Symposium on Visual Analytics Science and Technology, 2009. VAST 2009.*, pages 67 –74, oct. 2009.

[13] J. Choo, H. Lee, J. Kihm, and H. Park. iVisClassifier: An interactive visual analytics system for classification based on supervised dimension reduction. In *Visual Analytics Science and Technology (VAST), 2010 IEEE Conference on*, pages 27 –34, oct. 2010.

[14] J. Choo, H. Lee, Z. Liu, J. Stasko, and H. Park. An interactive visual testbed system for dimension reduction and clustering of large-scale high-dimensional data. pages 865402–865402–15, 2013.

[15] F. Chung. The heat kernel as the pagerank of a graph. *Proceedings of the National Academy of Sciences*, 104(50):19735–19740, 2007.

[16] E. Clarkson, K. Desai, and J. Foley. Resultmaps: Visualization for search interfaces. *Visualization and Computer Graphics, IEEE Transactions on*, 15(6):1057–1064, 2009.

[17] C. Dunne, B. Shneiderman, R. Gove, J. Klavans, and B. Dorr. Rapid understanding of scientific paper collections: Integrating statistics, text analytics, and visualization. *Journal of the American Society for Information Science and Technology*, 63(12):2351–2369, 2012.

[18] L. Eldén and H. Park. A procrustes problem on the stiefel manifold. *Numerische Mathematik*, 82:599–619, 1999.

[19] L. Good, A. Popat, W. Janssen, and E. Bier. A fluid interface for personal digital libraries. *Research and Advanced Technology for Digital Libraries*, pages 162–173, 2005.

[20] P. Howland and H. Park. Generalizing discriminant analysis using the generalized singular value decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(8):995 –1006, aug. 2004.

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

[22] N. Kampanya, R. Shen, S. Kim, C. North, and E. A. Fox. Citiviz: A visual user interface to the citidel system. *Research and Advanced Technology for Digital Libraries*, pages 122–133, 2004.

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

[24] J. Kim and H. Park. Sparse nonnegative matrix factorization for clustering. 2008.

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

[26] H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.

[27] H. Lee, J. Kihm, J. Choo, J. Stasko, and H. Park. iVisClustering: An interactive visual document clustering via topic modeling. *Computer Graphics Forum*, 31(3pt3):1155–1164, 2012.

[28] J. R. Lewis. Ibm computer usability satisfaction questionnaires: psychometric evaluation and instructions for use. *International Journal of Human-Computer Interaction*, 7(1):57–78, 1995.

[29] S. Liu, M. X. Zhou, S. Pan, Y. Song, W. Qian, W. Cai, and X. Lian. Tiara: Interactive, topic-based visual text summarization and analysis. *ACM Trans. Intell. Syst. Technol.*, 3(2):25:1–25:28, Feb. 2012.

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

[31] M. E. Newman. Fast algorithm for detecting community structure in networks. *Physical review E*, 69(6):066133, 2004.

[32] L. T. Nowell, R. K. France, D. Hix, L. S. Heath, and E. A. Fox. Visualizing search results: some alternatives to query-document similarity. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 67–75. ACM, 1996.

[33] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999. Previous number = SIDL-WP-1999-0120.

[34] P. Pirolli. Computational models of information scent-following in a very large browsable text collection. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 3–10. ACM, 1997.

[35] P. Pirolli and S. Card. Information foraging. *Psychological review*, 106(4):643–775, 1999.

[36] C. Plaisant. The challenge of information visualization evaluation. In *Proceedings of the working conference on Advanced visual interfaces*, pages 109–116. ACM, 2004.

[37] C. Plaisant, J.-D. Fekete, and G. Grinstein. Promoting insight-based evaluation of visualizations: From contest to benchmark repository. *Visualization and Computer Graphics, IEEE Transactions on*, 14(1):120–134, 2008.

[38] P. Saraiya, C. North, and K. Duca. An insight-based methodology for evaluating bioinformatics visualizations. *Visualization and Computer Graphics, IEEE Transactions on*, 11(4):443–456, 2005.

[39] M. M. Sebrechts, J. V. Cugini, S. J. Laskowski, J. Vasilakis, and M. S. Miller. Visualization of search results: a comparative evaluation of text, 2d, and 3d interfaces. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 3–10. ACM, 1999.

[40] R. Shen, N. S. Vemuri, W. Fan, R. da S Torres, and E. A. Fox. Exploring digital libraries: integrating browsing, searching, and visualization. In *Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries*, pages 1–10. ACM, 2006.

[41] J. Stasko, C. Görg, and Z. Liu. Jigsaw: supporting investigative analysis through interactive visualization. *Information Visualization*, 7(2):118–132, 2008.

[42] J. Stasko, C. Görg, and Z. Liu. Jigsaw: supporting investigative analysis through interactive visualization. *Information Visualization*, 7(2):118–132, Apr. 2008.

[43] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su. Arnetminer: extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '08, pages 990–998, New York, NY, USA, 2008. ACM.

[44] C. Wang and D. M. Blei. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 448–456. ACM, 2011.

[45] J. A. Wise, J. J. Thomas, K. Pennock, D. Lantrip, M. Pottier, A. Schur, and V. Crow. Visualizing the non-visual: Spatial analysis and interaction with information from text documents. In *Information Visualization, 1995. Proceedings.*, pages 51–58. IEEE, 1995.

[46] W. Xu, X. Liu, and Y. Gong. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, SIGIR '03, pages 267–273, New York, NY, USA, 2003. ACM.