Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Paper evolution graph: multi-view structural retrieval for academic literature*

Dan-ping LIAO, Yun-tao QIAN†‡

College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China

†E-mail: ytqian@zju.edu.cn

Received Feb. 13, 2017; Revision accepted July 20, 2017; Crosschecked Feb. 15, 2019

Abstract: Academic literature retrieval concerns about the selection of papers that are most likely to match a user's information needs. Most of the retrieval systems are limited to list-output models, in which the retrieval results are isolated from each other. In this paper, we aim to uncover the relationships between the retrieval results and propose a method to build structural retrieval results for academic literature, which we call a paper evolution graph (PEG). The PEG describes the evolution of diverse aspects of input queries through several evolution chains of papers. By using the author, citation, and content information, PEGs can uncover various underlying relationships among the papers and present the evolution of articles from multiple viewpoints. Our system supports three types of input queries: keyword query, single-paper query, and two-paper query. The construction of a PEG consists mainly of three steps. First, the papers are soft-clustered into communities via metagraph factorization, during which the topic distribution of each paper is obtained. Second, topically cohesive evolution chains are extracted from the communities that are relevant to the query. Each chain focuses on one aspect of the query. Finally, the extracted chains are combined to generate a PEG, which fully covers all the topics of the query. Experimental results on a real-world dataset demonstrate that the proposed method can construct meaningful PEGs.

Key words: Paper evolution graph; Academic literature retrieval; Metagraph factorization; Topic coherence
https://doi.org/10.1631/FITEE.1700105 **CLC number:** TP391

1 Introduction

Where did the idea of this paper come from? Are there any improved methods to do this? These are the questions beginners try to find answers when faced with an unfamiliar research territory. However, as the academic literature becomes ubiquitous, the problem of information overload has arisen. Users find an overwhelming number of publications that match their search queries but they can still be confused about where to start. For this reason, there is a growing need for techniques that can present the retrieved papers in a meaningful and effective way.

The existing academic literature retrieval systems, such as Google Scholar, Scopus, and Web of Science, play an important role in retrieving articles of interest. Applying advanced ranking algorithms, these systems can return articles that are most likely to match users' queries. However, although these systems are effective in retrieving relevant papers, the returned articles are displayed in a listed and isolated way. In other words, the underlying relationships between the retrieved articles remain unknown to users. It is still a problem for users to make a reading plan which guides them in deciding what to read first and next.

Some systems move beyond the list-output models and provide structural retrieval results. For example, Web of Science creates a citation map for

‡ Corresponding author

* Project supported by the National Key R&D Program of China (No. 2018YFB0505000) and the National Natural Science Foundation of China (No. 61571393)

ORCID: Yun-tao QIAN, http://orcid.org/0000-0002-7418-5891

each query paper based on its forward and backward citation relationships. However, when the query paper cites or is cited by a large number of papers, the papers are squeezed together such that it is hard to discern the papers. In addition, only the citation connection between the papers is elicited: there is no content/topic information presented. It still takes great effort for users to locate their papers of interest within such a big map.

In this study, we present the retrieved results in a way that explicitly shows the evolutionary relationship between the papers. As shown in Fig. 1, our system aims to string the retrieved articles together in an evolutional way and combine the strings to form a graph, which we call a paper evolution graph (PEG). Fig. 2 shows a simplified PEG. As can be seen, a PEG is a combination of several evolution chains depicted by different colors. Each evolution chain consists of a set of topically cohesive papers. Different chains focus on the evolution of the different topics relevant to the query. The common nodes of different chains reveal the intersection of different topics. For example, the PEG in Fig. 2 is generated based on an input query paper P. This PEG consists of three evolution chains describing the three technical routes that P involves.

A PEG allows users to browse the retrieved papers at a holistic level and navigate the overall aspects of the query. To fully uncover the relationships between academic articles, our system uses the content, author, and citation information to discover latent relationships between papers from multiple viewpoints. This allows our system to incorporate user preference to generate PEGs focusing on different types of coherence. For example, a PEG emphasizing author coherence is likely to consist of chains of articles published by the same authors, while a PEG emphasizing citation coherence tends to consist of articles that have citation relationships.

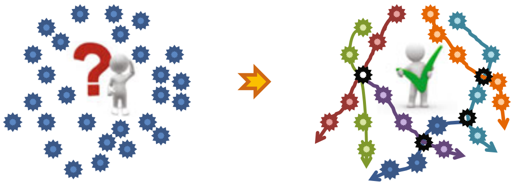


Fig. 1 Finding paths between pieces of messy information

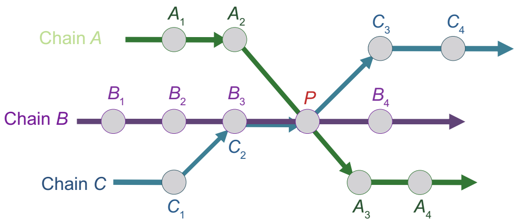


Fig. 2 A simplified example of a paper evolution graph

The process of building a PEG can be summarized in three steps. First, to fully cover the topics of a query paper in a PEG, we obtain the topic distribution of the papers in the dataset by multi-relational factorization of the metagraph of our system. After this step, articles in the dataset are soft-clustered into communities based on their topic distribution. Second, from each community that relates to the query, the most cohesive chain is extracted based on a proposed criterion for topic coherence. Finally, the extracted chains are combined to form a PEG.

To satisfy different user requirements, our system supports three types of queries: keyword query, single-paper query, and two-paper query.

1. Searching by keyword

Beginners who are new to an academic domain are always curious about the overall development of that domain. For example, a student who is new to "deep learning" would first search the words "deep learning" and then read a most classical article as suggested. Yet, what is the reader to do next? One option is to read a recently published paper to discover the latest development of deep learning. However, this beginner might have a problem in understanding the latest paper by jumping from the most basic theory to a much more sophisticated method; thus, he/she would have to read more articles to help digest the latest paper. Finding the requisite papers could be tedious. To help users achieve a comprehensive understanding of the domain, a PEG provides a graphic overview of the domain by depicting the relationships between the research branches and the development of each branch.

2. Searching by a single paper

The majority of research is done based on the previous studies. When reading a new paper, a beginner might want to find out how the idea of the paper was formed step by step from the very beginning, and whether there is any work that improves the technique in the paper. Our system can generate a PEG which explicitly shows the development of the query paper, not only making it easier for users to find former relevant papers, but also leading users to the later papers that are closely related to the query paper.

3. Searching by two papers

Sometimes users are interested in discovering the relationship between two papers. For example, one might want to find out the relationship between a paper that proposes a classical theory and a recent paper that uses a variant of the theory to solve a specific problem. Our system is able to present a PEG that shows a clear connection between the two papers, providing users with an idea of how the subject progressed step by step from the classical paper to the latest one. A PEG might also be useful for users who are curious about finding out the hidden connection between two papers that seem to be unrelated.

We believe that the PEG can serve as an effective tool to help users navigate unfamiliar territory and discover previously unknown relationships between articles. The main contributions of this paper are summarized as follows:

1. We propose the concept of paper evolution graph and formalize the criteria for evaluating evolution graphs.

2. We support three types of queries and provide efficient methods to construct evolution graphs given different types of queries.

3. We integrate user preferences into the framework to generate graphs describing the multi-view relationships among articles.

2 Related work

The problem of constructing a PEG relates to three aspects: document retrieval, organization of retrieval results, and topic discovery.

2.1 Document retrieval

The growing number of documents on the Web has accentuated the need for improving retrieval methods. The probability ranking principle (PRP) (Robertson, 1977) forms the bedrock of information retrieval. It aims to achieve the optimum retrieval by estimating the probability of relevance for each document (with respect to the current query) and ranking the documents according to the decreasing values of the probability of relevance. There is also a rise of use of language models in information retrieval (Lafferty and Zhai, 2001; Lavrenko and Croft, 2001). In the language modeling approach, each document is viewed as a language sample and a query is treated as a generation process. The retrieved documents are ranked according to the probabilities of generating a query from the corresponding language models of these documents. When the query has multiple interpretations, or there are multiple subtopics, systems are expected to balance relevance and diversity (Chen and Karger, 2006; Agrawal et al., 2009). The basic premise of result diversification is that the relevance of a set of documents depends not only on the individual relevance of its members, but also on how they relate to each other. Maximizing diversity is especially useful in the feedback-relevant retrieval systems and commercial websites (Shen and Zhai, 2005; Yu et al., 2014).

When it comes to academic literature, a paper's citation count is widely used in evaluating the importance of a paper since it has been shown to strongly correlate with academic literature impact (Narin, 1976). The Thomson Scientific Institute for Scientific Information Impact Factor (ISI IF) is a representative approach using a paper's citation count, and is defined as the mean number of citations to articles published in a journal over a two-year period (Garfield, 1979). However, citation counting has well-known limitations: citing papers with a high impact and ones with a low impact are treated equally in standard citation counting. Google's PageRank algorithm counts not only the number of hyperlinks to a page. It also computes the status of a web page based on a combination of the number of hyperlinks that point to the page and the status of the pages from which the hyperlinks originate (Brin and Page, 1998). Papers with more citations are generally ranked higher, and they get a further boost if they are referenced by highly cited articles (Butler, 2004). Chen et al. (2007) applied the PageRank algorithm to the scientific citation networks. They found out that according to the PageRank model, although some classical articles in the physics domain have a small number of citations, they have a very high PageRank.

2.2 Retrieval result organization

Search engines and recommendation systems play a crucial role in paper retrieval. However, most of them are limited to list-output models; i.e., the retrieval results are listed one by one and isolated from each other. Although these systems display useful information, simply listing the output is not sufficient for users to capture the relationships among retrieval results. There are a few systems that move beyond the list-output model. In the topic detection task, Jo et al. (2011) aimed to discover the evolution of topics over time in paper collection. The discovered topics were connected to form a topic evolution graph using a measure derived from the underlying paper network. Graph- and network-based models were also used to represent and analyze the relationships among scientific authors. For example, Newman (2001) and Tang et al. (2008) used the publications of authors to analyze and visualize co-author and citation relationships in scientific literature.

Representing the retrieval results in a structured way has attracted more attention beyond the academic literature retrieval domain. The ostensive browsing model (Campbell, 2000) uses paths and nodes to represent the interactive feedback-relevant searching process, where users move from node (information object) to node via links (accessibility relationships). The path is a sequence of nodes for users to trace and explore. In the news analysis domain, numerous works have proposed different notions of storylines (Allan et al., 2001; Shahaf and Guestrin, 2010; Ahmed et al., 2011; Yan et al., 2011). In addition, graph representations are common across a variety of related problems. For example, Kleinberg (2003) focused on discovering bursty and hierarchical structures in text streams. Makkonen (2003) suggested modeling news topics in terms of their evolving events. Mei and Zhai (2005) aimed to discover and summarize the evolutionary patterns of themes in a text stream.

In a work most related to ours, Nallapati et al. (2004) proposed a process called event threading, in which the structure of news events and their dependencies in a news topic were captured to generate a graph structure. Shahaf et al. (2012) created an evolution map for news events. They proposed the notion of topic coherence for an evolution path. Our work shares the same advantages of the previous works in that we try to uncover the relationships between retrieval results. However, we focus on the academic domain, where more information such as authors and citations, rather than just the content information, can be used.

2.3 Topic discovery

Due to its importance and great application potential, topic research in scientific literature has recently attracted rapidly growing interest (Mei et al., 2006; Schult and Spiliopoulou, 2006; Spiliopoulou et al., 2006). Many existing approaches for scientific literature topic detection model a paper as a bag of words (Bolelli et al., 2009; Gohr et al., 2009). However, the bag-of-words model is effective for only discovering topics when papers share a large proportion of lexically equivalent terms. Several studies integrated author information and content information to help detect topics (Rosen-Zvi et al., 2004; Steyvers et al., 2004; Zhou et al., 2006). He et al. (2009) addressed the problem of topic detection by adapting the latent Dirichlet allocation model (Blei et al., 2003) to the citation network. In Small (1973), the relationship between two papers was measured via their co-citations. In this study, we integrate the content, citation, and author information to discover academic topics from different viewpoints.

3 Overview of paper evolution graph construction

In this section, we present the outline of the proposed PEG construction. First, we give definitions of paper evolution chain and paper evolution graph.

Definition 1 (Paper evolution chain) A paper evolution chain L of length n is a simple directed path with n vertices, denoted by $L = (p_1, p_2, \ldots, p_n)$, where $\{p_i\}_{i=1}^{n}$ is a sequence of chronologically ordered and topically cohesive papers.

Definition 2 (Paper evolution graph) A paper evolution graph $G = (V, E)$ is a directed graph consisting of several evolution chains L_i, denoted by $G = U(L_i)$, where each L_i focuses on different topics.

Fig. 3 gives the procedure for constructing a PEG.

In the first step, we build the metagraph of our system specifying the relationships between “word,” “author,” and “paper.” A metagraph is a relational hypergraph representing multi-relational and multi-dimensional data (Lin et al., 2009). It is a graph with its nodes (called facets) representing the entities and its edges (called hyperedges) corresponding to the interactions between the nodes. A metagragh is different from the traditional graph in that each node/facet represents an ensemble of the entity. For example, the author facet is a set of authors and the paper facet is a set of papers. The hyperedge connecting the facets represents the relationship between the entity sets. Fig. 4 shows the metagraph of our system.

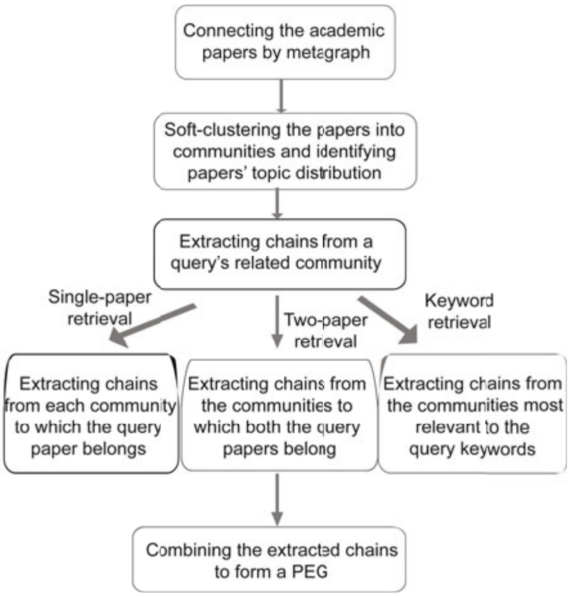

Connecting the academic papers by metagraph
Soft-clustering the papers into communities and identifying papers' topic distribution
Extracting chains from a query's related community
Single-paper retrieval
Two-paper retrieval
Keyword retrieval
Extracting chains from each community to which the query paper belongs
Extracting chains from the communities to which both the query papers belong
Extracting chains from the communities most relevant to the query keywords
Combining the extracted chains to form a PEG

Fig. 3 Framework of the paper evolution graph (PEG) construction approach

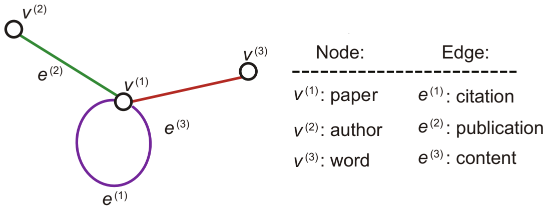


Fig. 4 Metagraph of our system

We prepare three types of data according to three relationships in the metagraph: the "content" relationship between paper facet and word facet, the "publication" relationship between paper facet and author facet, and the "citation" relationship between paper facet and paper facet. Each relationship corresponds to an observed data.

The second step is to soft-cluster the papers in the dataset into communities according to the papers' topic distribution, which is achieved by multi-relational factorization of the metagraph. In this step, each paper can be assigned to one community or more communities.

The third step is to extract topically cohesive chains from the communities that are relevant to the query. In this step, we first define the topic coherence of a given chain of papers. Then the most coherent chains are extracted from each of the query-relevant community.

For different types of queries, the definition of query-relevant community varies slightly. For a single-paper query, the relevant communities are defined as the communities consisting of the query paper. For a two-paper query, the relevant communities are defined as the communities consisting both of the query papers. For a keyword query, the query-relevant communities are the communities that include the papers relevant to the keyword.

After the most coherent evolution chains are extracted from the relevant communities, these chains are combined to form a PEG in the last step. Since each chain focuses on one aspect of the query, combining the chains gives us a comprehensive and holistic view of evolution of the query.

4 Identifying the topic distribution of papers

To construct a PEG which fully covers the topics of a query, we should first identify the topic distribution of the papers in the dataset. In this section, we introduce the approach to obtain the topic distribution of papers via metagraph factorization. The first step is to build a metagraph that covers the papers' content, authorship, and citation information. Then the topic distribution of each paper is obtained by metagraph factorization.

4.1 Constructing a metagraph

In this study, each paper is modeled as a probabilistic mixture of topics; i.e., a paper belongs to one topic or more topics with different probabilities. The topic distribution of a paper is defined as follows:

Definition 3 (Topic distribution) The topic distribution of a paper p is a nonnegative vector $\boldsymbol{T} = (T_1, T_2, \ldots, T_C)$, where C is the number of topics, T_i is the probability that paper p belongs to the i^{th} topic, and $\sum_i T_i = 1$.

Fig. 5 shows an example of topic distribution, where the x axis denotes the index of topics and the y axis is the probability that a paper belongs to a specific topic. In this figure, $T_8 = 0.26$ and

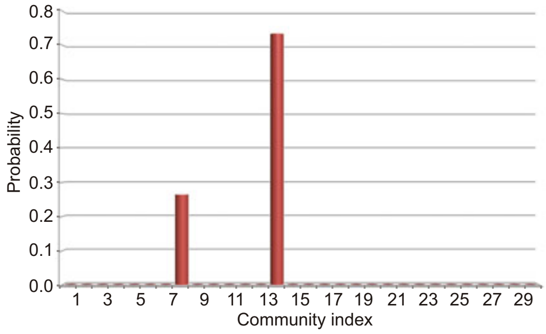


Fig. 5 An example of topic distribution

$T_{14} = 0.74$.

In our approach, topic distribution is calculated by metagraph factorization based clustering, in which similar articles are grouped to the same topics. Three types of papers are likely to share similar topics: papers that are similar in content, papers that share the same authors, and papers that have citation relationships. To fully uncover the relationships among papers, we use three types of article information in the clustering step:

1. Content information

The content of a paper conveys its topic in a most direct way. Papers with a high content similarity (e.g., word vector based similarity) will have similar topics.

2. Authorship information

Since the research interests of a researcher are limited, papers published by the same author are likely to focus on the same topic.

3. Citation information

If a paper cites another paper, there is a high probability that the two papers have the same topic.

The relationships between papers can be measured in the paper-word space, paper-author space, and paper-paper space. The three spaces are not independent. In fact, they share one common dimension: the paper dimension. The relationships between the spaces can be represented by a metagraph (Fig. 4).

Let V and R denote the sets of facets and edges respectively, where $v^{(i)}$ denotes the i^{th} facet and $e^{(r)}$ represents the r^{th} edge. A hyperedge/relation $e^{(r)}$ is said to be incident to a facet $v^{(q)}$ if $v^{(q)} \in e^{(r)}$. There are three facets $V = \{v^{(i)}\}$ $(i = 1, 2, 3)$ and three relationships $E = \{e^{(j)}\}$ $(j = 1, 2, 3)$ in Fig. 4. The facets (paper, author, and word) are connected by hyperedges $e^{(1)}$, $e^{(2)}$, and $e^{(3)}$. Thus, $e^{(1)} = \{v^{(1)}, v^{(1)}\}$ represents the citation relationship between papers; $e^{(2)} = \{v^{(1)}, v^{(2)}\}$ represents the content relationship between papers and words; and $e^{(3)} = \{v^{(1)}, v^{(3)}\}$ represents the publishing relationship between authors and papers.

By constructing the metagraph, we can integrate the content, author, and citation information, which were originally independent.

4.2 Soft-clustering papers into communities

A clustering technique is crucial for large-scale topic discovery. Since there are three relationships in our system, a multi-relational clustering technique is required. Clustering entities with multiple relationships consider joint factorization over two or more matrices. There are numerous studies addressing multi-relational clustering (Long et al., 2006; Banerjee et al., 2007; Zhu et al., 2007; Lin et al., 2009). In this study, we apply a metagraph factorization method proposed by Lin et al. (2009) to soft-cluster the papers. This approach is a fast and practical approach to extract communities from multiple relationships on the basis of tensor operation. Furthermore, the approach is incremental in that it can be readily modified to deal with time evolving data, where the relational data are modeled as evolving tensor sequences, which makes the technique scalable.

4.2.1 Background knowledge of tensors

In this subsection, we provide the background knowledge on the tensor and the operations used in this study. A tensor is a mathematical representation of a multi-way array. The order of a tensor is the number of modes (or ways). For example, a first-order tensor is a vector, a second-order tensor is a matrix, and a third-order tensor is a cube. In this study, we use $\boldsymbol{x}$ as a vector, $\boldsymbol{X}$ as a matrix, and $\mathcal{X}$ as a tensor. The dimensionality of a mode is the number of elements in that mode. For example, a nonnegative tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ has three modes with dimensionalities of I_1, I_2, and I_3, respectively. Tensor factorization or multi-linear matrix factorization is widely used in recommender systems (Yu et al., 2015). Five basic tensor operations are used in this study:

1. Mode-d unfolding

Unfolding is a process of reordering the elements of an M-way array into a matrix. The mode-d unfolding of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is denoted by $\boldsymbol{X}_{(d)}$; i.e., $\text{unfold}(\mathcal{X}, d) = \boldsymbol{X}_{(d)} \in \mathbb{R}^{I_d \times \prod_{q \in 1,2,\ldots,M, q \neq d} I_q}$. Unfolding a tensor on mode d returns a matrix with I_d rows. Its column number is the product of dimensionalities of all the modes except mode d. The inverse operation is denoted as $\mathcal{X} = \text{fold}(\boldsymbol{X}_{(d)}) \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$. Unfolding can be defined on two or more modes. For example, unfolding a tensor along with modes c and d, is defined by $\text{unfold}(\mathcal{X}, (c,d)) = \boldsymbol{X}_{(c,d)} \in \mathbb{R}^{I_c \times I_d \times \prod_{q \in 1 \cdots M, q \neq c,d} I_q}$, where $\boldsymbol{X}_{(c,d)}$ is a three-way tensor (a cube).

2. Mode-d product

The mode-d product $\mathcal{Y} = \mathcal{X} \times_d \boldsymbol{A}$ of a tensor $\mathcal{X} \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_N}$ and a matrix $\boldsymbol{A} \in \mathbb{R}^{I_n \times J_n}$ is a tensor $\mathcal{Y} \in \mathbb{R}^{J_1 \times \cdots \times J_{n-1} \times I_n \times J_{n+1} \times \cdots \times J_N}$. Elementwise, we have $\mathcal{Y}_{j_1, j_2, \cdots, j_{n-1}, i_n, j_{n+1}, \cdots, j_N} = \sum_{j_n=1}^{J_n} g_{j_1, j_2, \cdots, j_N} a_{i_n, j_n}$.

3. Tensor vectorization

Vectorization is the process of linearizing the elements of an M-mode array into a vector, and is denoted by $\boldsymbol{x} = \text{vec}(\mathcal{X})$.

4. Khatri-Rao product

For two matrices $\boldsymbol{A} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_J] \in \mathbb{R}^{I \times J}$ and $\boldsymbol{B} = [\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_J] \in \mathbb{R}^{T \times J}$ with the same number of columns J, their Khatri-Rao product, denoted by "$\odot$," is defined by $\boldsymbol{A} \odot \boldsymbol{B} = [\text{vec}(\boldsymbol{b}_1 \boldsymbol{a}_1^{\mathrm{T}}),\ \text{vec}(\boldsymbol{b}_2 \boldsymbol{a}_2^{\mathrm{T}}), \ldots,\ \text{vec}(\boldsymbol{b}_J \boldsymbol{a}_J^{\mathrm{T}})] \in \mathbb{R}^{IT \times J}$.

5. Mode-d accumulation

A mode-d accumulation of a tensor $\mathcal{X}$ is defined as $\text{acc}(\mathcal{X}, d) = \boldsymbol{X}_{(d)} \boldsymbol{1} \in \mathbb{R}^{I_d}$.

Accumulating a tensor on mode d can be calculated by unfolding the tensor on mode d into a matrix and then multiplying the matrix with an all-one vector (summing up all the columns). Accumulation on two modes c and d is defined by $\text{acc}(\mathcal{X}, (c,d)) = \boldsymbol{X}_{(c,d)} \times_3 \boldsymbol{1} \in \mathbb{R}^{I_c \times I_d}$. Readers can refer to Bader and Kolda (2006) for a more comprehensive review of tensors.

4.2.2 Clustering papers into communities

Since the hyperedges in our system are all two-way relationships, a set of observations is represented as a two-way tensor, i.e., a matrix. Each relation $e^{(j)}$ forms a matrix and corresponds to an observed data. The "citation" hyperedge corresponds to a second-order tensor $\mathcal{X}^{(1)} \in \mathbb{R}^{P \times P}$, where P is the number of papers. The "content" hyperedge corresponds to a second-order tensor $\mathcal{X}^{(2)} \in \mathbb{R}^{P \times W}$, where W is the number of words. The "author" hyperedge corresponds to a second-order tensor $\mathcal{X}^{(3)} \in \mathbb{R}^{P \times N}$, where N is the number of authors.

The relationship between any two members i and j in community k is denoted as x_{ij}. Let $p_{k \to i}$ indicate how likely an interaction in the k^{th} community involves the i^{th} member and p_k the probability of the interaction in the k^{th} community. x_{ij} can be represented by $x_{ij} \approx \sum_k p_k \cdot p_{k \to i} \cdot p_{k \to j}$. A set of such relationships among entities in facets $v^{(a)}$ and $v^{(b)}$ can be written by

$$\mathcal{X} \approx \sum_{k=1}^{K} p_k \circ u_k^{(a)} \circ u_k^{(b)} = \boldsymbol{P} \times_1 \boldsymbol{U}^{(a)} \times_2 \boldsymbol{U}^{(b)}. \quad (1)$$

The data tensor $\mathcal{X} \in \mathbb{R}_+^{I_a \times I_b}$ represents the observed two-way interactions among facets $v^{(a)}$ and $v^{(b)}$, and K is the number of communities. $\boldsymbol{U}^{(q)}$ is an $I_q \times K$ matrix, where I_q is the size of $v^{(q)}$. $p_{k \to i_q}$ is the (i_q, k)-element of $\boldsymbol{U}^{(q)}$ for $q = a, b$. $\boldsymbol{P}$ is a diagonal matrix with the diagonal elements representing the probabilities of each community, i.e., $p_k = \boldsymbol{P}(k,k)$.

Eq. (1) can be viewed as community discovery in a single relation. Since there are three relations in our system, our objective is to factorize all the data tensors such that all tensors can be approximated by a common nonnegative core tensor $\boldsymbol{P}$ and a shared nonnegative factor $\boldsymbol{U}^{(1)}$, i.e., to minimize the following cost function:

$$\begin{aligned} J(G) = \min_{z, U^{(q)}} & \; w_1 D(\mathcal{X}^{(1)} \| \boldsymbol{P} \times_1 \boldsymbol{U}^{(1)} \times_2 \boldsymbol{U}^{(1)}) \\ & + \; w_2 D(\mathcal{X}^{(2)} \| \boldsymbol{P} \times_1 \boldsymbol{U}^{(1)} \times_2 \boldsymbol{U}^{(2)}) \\ & + \; w_3 D(\mathcal{X}^{(3)} \| \boldsymbol{P} \times_1 \boldsymbol{U}^{(1)} \times_2 \boldsymbol{U}^{(3)}) \qquad (2) \\ \text{s.t.} \quad & \boldsymbol{P} \in \mathbb{R}_+^{K \times K}, \boldsymbol{U}^{(q)} \in \mathbb{R}_+^{I_q \times K}, \ \forall\ q, \\ & \sum_i \boldsymbol{U}_{ik}^{(q)} = 1, \ \forall\ q, \ \forall\ k, \end{aligned}$$

where $D(\cdot \| \cdot)$ is the KL-divergence and w_r ($r = 1,\ 2,\ 3$) is the weight of $\mathcal{X}^{(r)}$.

We apply the tensor operation based metagraph factorization algorithm developed in Lin et al. (2009) to find a local minimum solution. The solution shares the same form of the expectation-maximization algorithm and can be found by the following multiplicative updating algorithm:

In the first step, for each $e^{(r)}$, compute a tensor

$\mathcal{C}^{(r)} \in \mathbb{R}_{+}^{I_1^r \times I_2^r \times K}$ by

$$\mu^{(r)} \leftarrow \mathrm{vec}(\mathcal{X}^{(r)} \oslash (\boldsymbol{P} \prod_{n:v^{(n)} \in e^{(r)}} \times_n \boldsymbol{U}^{(n)})), \qquad (3)$$

$$\mathcal{C}^{(r)} = \mathrm{fold}(\mu^{(r)} \odot (\boldsymbol{P} \odot \boldsymbol{U}^{(2)}_{(r)} \odot \boldsymbol{U}^{(1)}_{(r)})^{\mathrm{T}}), \qquad (4)$$

where "$\oslash$" is the elementwise division.

In the second step, $\boldsymbol{P}$ and $\boldsymbol{U}^{(q)}$ are updated respectively by

$$\boldsymbol{P} \leftarrow \frac{1}{3} \sum_{r \in E} \mathrm{acc}(\mathcal{C}^{(r)}, 3), \qquad (5)$$

$$\boldsymbol{U}^{(q)} \leftarrow \sum_{l:e^{(l)} \in v^{(q)}} \mathrm{acc}(\mathcal{C}^{(l)}, (3, q)). \qquad (6)$$

Eqs. (3) and (4) correspond to the E-step and Eqs. (5) and (6) correspond to the M-step. The information in each data tensor is aggregated at the E-step and is shared by the core tensor and all facet factors at the M-step. Algorithm 1 summarizes the process of metagraph factorization.

Algorithm 1 Metagraph factorization

```
Input: metagraph $G = (V, E)$ and data tensors $\mathcal{X}^{(1)}$, $\mathcal{X}^{(2)}$, and $\mathcal{X}^{(3)}$ on $G$.
Output: $\boldsymbol{P}$, $\boldsymbol{U}^{(1)}$, $\boldsymbol{U}^{(2)}$, and $\boldsymbol{U}^{(3)}$.
Initialize $\boldsymbol{P}$, $\boldsymbol{U}^{(1)}$, $\boldsymbol{U}^{(2)}$, and $\boldsymbol{U}^{(3)}$; Repeat until convergence
for each $r \in E$ do
    Compute $\mathcal{C}^{(r)}$ by Eqs. (3) and (4)
end for
Update $\boldsymbol{P}$ by Eq. (5)
for each $q \in V$ do
    Update $\boldsymbol{U}^{(1)}$, $\boldsymbol{U}^{(2)}$, and $\boldsymbol{U}^{(3)}$ by Eq. (6)
end for
```

After multi-relational factorization, the topic distribution $T = (p(1|i), \ldots, p(k|i), \ldots, p(K|i))$ of the i^{th} paper in the dataset is calculated by $p(k|i) = p(i|k)p(k)/p(i)$. Here, $p(i|k)$ is the (i, k)-element of $U^{(1)}$, which denotes how likely the k^{th} community includes the i^{th} paper. $p(i)$ is the probability of a relation involving paper i and is defined as $p(i) = \sum_k p(i|k)p(k)$.

After the topic distribution is acquired, each paper is assigned to one or more (usually 1 to 3) communities, which describe different topics of the paper. The i^{th} paper in the dataset is considered to belong to community k if $p(k|i) \geq \mathrm{Com}_t$, where Com_t is a threshold parameter ($\mathrm{Com}_t = 0.2$ in this study). For example, the paper with a topic distribution shown in Fig. 5 belongs to community 8 and community 14. Users can tune down Com_t to retrieve papers from more communities when a PEG with more diversity is desired.

5 Generating evolution graphs

In this section, we describe our approach for extracting topically cohesive chains and constructing PEGs. To extract cohesive chains, we first define the link strength between two papers, based on which we define the topic coherence of a chain. Then we extract chains with the most coherent topic from the query-related communities. After topically cohesive chains are extracted, they are combined to construct a PEG.

5.1 Computing link strength between adjacent papers

Given a chain of papers $L = (p_1, p_2, \ldots, p_n)$, the link strength between adjacent papers can be measured by the similarity between the two papers. Traditionally, papers are represented by vectors of term frequencies, i.e., using the "bag-of-words" model. Each term is then assigned a "weight of importance" using a weighting metric such as the TF-IDF weighting scheme (Salton, 1971). A simple measurement of the link strength is the word vector based similarity between papers, e.g., the cosine distance between word vectors. However, this type of similarity is not always informative for measuring the link strength. First, since the bag-of-words consists only of terms that appear in a paper's original source text, there exist two linguistic phenomena: ambiguity and synonymy (Aljaber et al., 2010). Ambiguity occurs when papers share lexically similar, but semantically distinct terms. It can make papers appear more similar than they actually are. Synonymy, on the other hand, occurs when two papers share semantically related, but lexically dissimilar words. As a result, two correlative papers appear less correlative than they actually are. Second, word vector based similarity always considers each term to be of equal importance. However, when calculating similarity under a certain topic, some terms are more significant than other terms. Say, there is a paper focusing on the non-negative matrix factorization (NMF) and its application to "image compression." Another paper also applies the NMF method, but uses it to address a data mining problem. The two papers are relevant because they both use NMF to solve problems. However, when the topic of a chain is image compression, these two papers should not be considered relevant, which means we should give a small weight to the term "NMF" when calculating paper similarity under the image compression topic. In this study, we consider the word influence in similarity calculations, i.e., the influence of a word w in the relation of p_i and p_{i+1}. With word influence, the similarity between two papers can be obtained under different topics by assigning different weights to each word.

Among the several proposed methods for measuring word influence, the majority of them focus on directed weighted graphs (e.g., the web, social networks, and citations), in which influence is considered to spread through the edges. Methods such as PageRank (Brin and Page, 1998), authority computation (Kleinberg, 1999), and random graph simulations (Kempe et al., 2003) all use the link structure. In this study, we use the influence calculating algorithm proposed by Shahaf and Guestrin (2010), where word influence was obtained by a random walk. The algorithm overcomes the two drawbacks of word vector based similarity. It creates a network between all the papers and words, where the relation between two papers is acquired by the word influence propagating in the network. This framework allows two papers to be linked through a word w even if w does not appear in the two articles.

To create the network, we first create a bipartite directed graph $B = (V, E)$. The vertices $V = V_P \bigcup V_W$ correspond to papers and words. Fig. 6a shows a simple bipartite graph: the squares on the top represent four papers and the circles on the bottom denote five words. We add both edges (w, p) and (p, w) for each pair of word w and paper p. The weights of edges indicate the strength of the relevance between papers and words.

For each paper, the weights of the paper-to-word edges are assigned with their TF-IDF (term frequency-inverse document frequency) features. Since the weights are considered to be random walk probabilities, they are normalized over all the words such that $\sum_i \text{weight}(p, w_i) = 1$. The word-to-paper weights are initialized in the same way as the paper-to-word edges, but normalized over the papers.

To calculate influence$(p_i, p_j|w_k)$, we first compute the stationary distribution for random walks starting from p_i. The probability $p(p_j|p_i)$ can be calculated from p_i to p_j through the whole word set. To specify the effect of w_k on these walks, a graph B' which is the same as B is constructed except that there is no way out of the node w_k; i.e., the weights of the word-to-paper edges of w_k are set to 0. Again, the stationary distribution on B' is calculated starting from p_i. This time the probability $p(p_j|p_i)$ is computed without the influence of w_k. We denote this probability as $p_{-w_k}(p_j|p_i)$. The word influence w_k between p_i and p_j is defined as the probability difference between the two distributions and can be calculated by

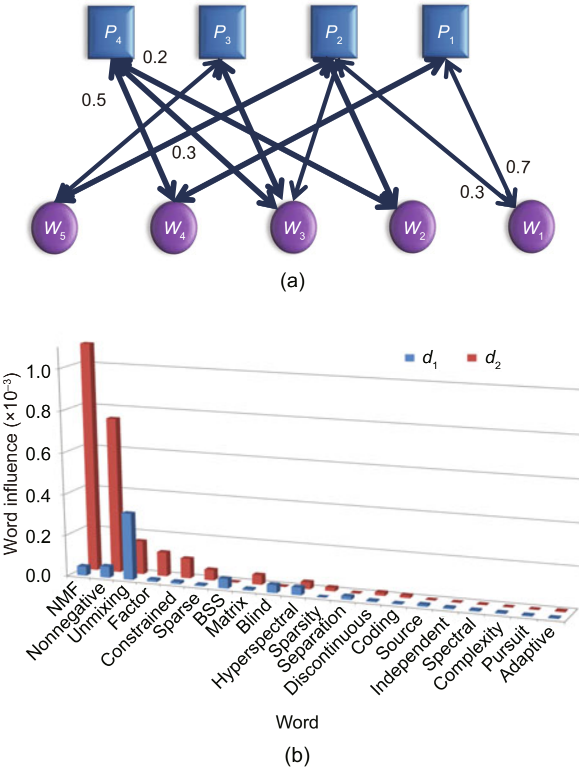


Fig. 6 Word influence between papers: (a) bipartite graph of papers and words; (b) word influence between d and d_1, d and d_2. Blue bars represent word influence for d and d_1 while red bars denote the word influence for d and d_2. References to color refer to the online version of this figure

$$\text{influence}(p_i, p_j|w) = p(p_j|p_i) - p_{-w_k}(p_j|p_i). \quad (7)$$

Fig. 6b shows an example of the word influence between paper d and two other articles d_1 and d_2.

d: constrained non-negative matrix factorization for hyperspectral unmixing;

d_1: spectral and spatial complexity-based hyperspectral unmixing;

d_2: hyperspectral unmixing via $L_{1/2}$ sparsity constrained non-negative matrix factorization.

We can see that words such as "unmixing," "BSS," "blind," and "hyperspectral" have higher influences in the relation between d and d_1, while "NMF," "non-negative," and "sparse" have higher influence in the relation between d and d_2.

After obtaining the word influence between each pair of papers in the dataset, the similarity between two papers can be computed under different topics $\boldsymbol{T}$ by assigning different weights to each word. Topic $\boldsymbol{T}_k$ is described by vector $\boldsymbol{T}_k = [t_1^k, \ldots, t_n^k, \ldots, t_W^k]$, where t_n^k is the weight of word w_n and W is the number of words. The similarity between two papers under a certain topic can be formulated as the sum of influences through all the words:

$$\begin{aligned} \mathrm{sim}(p_i, p_j, \boldsymbol{T}_k) &= \sum_n t_n^k \cdot \mathrm{influence}(p_i, p_j | w_n) \\ \text{s.t.} \ & \sum_n t_n^k = 1. \end{aligned} \tag{8}$$

5.2 Evaluating coherence of chains

In this study, the coherence of the chain is measured by the strength of its weakest link due to the fact that a single poor transition can destroy the coherence of the entire chain (Shahaf and Guestrin, 2010). Given a chain of papers $L = (p_1, p_2, \ldots, p_m)$, its coherence is defined as the maximum value of its weakest link strength among all the possible topics:

$$\mathrm{cohere}(L) = \max_{\boldsymbol{T}_k} \min_{j=1,2,\ldots,m-1} \mathrm{sim}(p_j, p_{j+1}, \boldsymbol{T}_k). \tag{9}$$

To determine the topic coherence of a chain, we need to find a topic that maximizes the optimization problem (9). However, problem (9) considers the topic of a chain to be constant. In other words, the importance of the words is fixed while computing the link strength between adjacent papers. In fact, the research topic is always gradually changing over time. A more reasonable objective function would consider the adjacent papers' similarity to be computed under a set of smoothly evolving topics $M = (\boldsymbol{T}_1, \boldsymbol{T}_2, \ldots, \boldsymbol{T}_{m-1})$. In this case, the coherence of a chain is defined as

$$\begin{aligned} \mathrm{cohere}(L) &= \max_M \min_{j=1,2,\ldots,m-1} \mathrm{sim}(p_j, p_{j+1}, \boldsymbol{T}_j) \\ &\text{s.t.} \ \sum_i t_i^j = 1, \|t_i^j - t_i^{j+1}\| \le r, \end{aligned} \tag{10}$$

where t_i^j is the weight of the i^{th} word in topic $\boldsymbol{T}_j$, which is used to calculate the similarity between p_j and p_{j+1}. By introducing the range parameter r, the word importance (the topic) is allowed to change slightly between adjacent pairs of papers along the chain. Our goal is to find a set of gradually changed topics such that the chain reaches its highest coherence. Eq. (10) can be readily formalized as a linear programming problem. Since the length of the chain and the number of total words are limited, the linear programming problem can be solved fast.

After selecting the most topically cohesive chains, these chains are combined to construct a PEG. Since a paper can belong to k ($k >= 1$) communities after soft-clustering, chains extracted from different communities may share common nodes. Fig. 7 illustrates the process of combining two chains to form a graph. When chains share the same nodes, the common nodes are merged. The common nodes uncover the intersection between different technique routes.

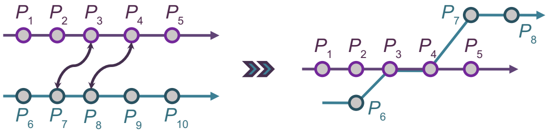


Fig. 7 Combining chains to construct a graph

5.3 Chain extraction for three types of queries

Our system supports three types of input queries: keyword, single-paper, and two-paper queries. For different types of queries, the PEG can be generated under the same framework with small variance.

Assume that the length of the evolution chain is set to n either by our system or by the user. When the query is a single paper p, our model generates a PEG discovering the evolution of p's topics. The first step is to find the communities that include p according to its topic distribution. Then from each of p's communities, our model selects a chain that has the strongest topic coherence from all the possible chains that contain p as one of its nodes. Finally, the chains selected from different communities are combined to form a PEG.

For two query papers p_s and p_t, our model constructs a PEG that uncovers the evolution relation between the papers. Assume that p_s was published before p_t. The first step is to find the shared communities of p_s and p_t according to their topic distribution. Let C_i $(i = 1, 2, \ldots, R)$ denote the i^{th} common community of p_s and p_t, where R is the number of shared communities. For each C_i, our model computes the coherence of all the chains that start with p_s and end with p_t using Eq. (10). Then a chain with the largest coherence is extracted from each C_i. These chains are then combined to construct a PEG.

When the query is a keyword, our system generates a PEG consisting of papers that are most relevant to the keyword using the following steps:

1. Select N papers that are most relevant to the query keyword by the TF-IDF feature of the papers.

2. Divide the selected papers into groups according to the communities they belong to; i.e., papers that belong to the same community are assigned to the same group.

3. In each group, compute the topic coherence of all the possible chains of length n and choose the most coherent chain.

4. Combine the chains to form a PEG.

5.4 Some issues on implementation

Two of the most time-consuming parts of the proposed model are: (1) hypergraph factorization to generate communities, and (2) chain extraction from each query-related community.

Hypergraph factorization has a time complexity of $O(n)$ per iteration. The factorization can be readily modified to deal with time evolving data as proposed by Lin et al. (2009), where the relational data were modeled as evolving tensor sequences, making the technique well scalable.

Our method aims to search for the most coherent chains from each query-relevant community. Since there are an enormous number of possible chains in a community, an exhaustive search method is not feasible when the dataset is very large. To accelerate computation, we narrow down the search region by selecting a small number of the most relevant papers as the candidates for constructing chains.

When the query is a single paper p, we select M papers that are most similar to p in each community that p belongs to, using the similarity defined as

$$R(p, p_m) = \sum_k P(k)P(k \to p)P(k \to p_m), \quad (11)$$

where k is a community and the summation is taken over all the communities.

For two query papers p_s and p_t, M papers that are most similar to the two papers are selected in each community to which both p_s and p_t belong, using the similarity defined as

$$\begin{aligned} R(p_s, p_t, p_m) = \sum_k & P(k)P(k \to p_m) \\ & \cdot P(k \to p_s)P(k \to p_t). \end{aligned} \quad (12)$$

In our experiment, M is set to 50.

Chain extraction involves candidate paper selection and linear programming (Eq. (10)) to determine the coherence of an evolution chain. The time complexity of candidate paper selection is $O(n)$. Since the length of the chain and the number of total words are limited, the linear programming problem can be solved fast.

To discover whether the acceleration process will affect the final retrieval results, we conducted two sets of experiments, where one set was done with acceleration and the other without acceleration. In most cases the two groups of results were exactly the same, which means that the papers ruled out from the candidates do not contribute to the final results.

6 Experiments

A real-world dataset of 24 491 papers was collected to test the effectiveness of our structural retrieval approach. Three types of queries were conducted on the dataset. In addition, user preference was incorporated into our system to better meet users' needs.

6.1 Dataset description

We crawled and parsed the articles from the journal called *IEEE Transactions on Geoscience and Remote Sensing* (TGRS) for the years 1980–2012 and a conference called IEEE International Geoscience and Remote Sensing Symposium (IGARSS) for the years 1988–2012. We chose this dataset because remote sensing is one of the research domain of our laboratory, which allows us to better analyze the retrieval results. Each of the three relations ("content," "author," and "citation") in the dataset corresponds to a second-order tensor (a matrix). The relations are summarized in Table 1. The "content" data measures the relationships between papers and words, where the words are extracted from the title, keywords, and abstract of each paper with stop words removed and stemming. The TF-IDF metric is used to measure how important a word is to a paper. For paper p, its TF-IDF feature $\boldsymbol{\delta}^p = [\delta_1^p, \delta_2^p, \ldots, \delta_N^p]$ is calculated by

$$\delta_i^p = \mathrm{tf}_{w_i,p} \log \frac{|P|}{\{p' \in P | w_i \in p'\}}, \tag{13}$$

where $|P|$ is the number of papers in the dataset and $\mathrm{tf}_{w_i,p}$ is the frequency of word w_i in p. In this study, we use the number of times that word w_i occurs in document p as the word frequency. The "author" data are a "0-1" matrix with "1" referring to an author publishing a paper. The "citation" data are also a "0-1" matrix where "1" refers to a paper citing another paper.

Table 1 Summary of the relations in the TGRS dataset

Relation	Tensor (incident facets)	Size
Content	Paper, word	24 491 × 27 730
Author	Paper, author	24 491 × 38 094
Citation	Paper, paper	24 491 × 24 491

At the metagraph factorization stage, papers in the dataset were clustered into 30 communities. Four of the communities are displayed as word clouds in Fig. 8. The three relations are assigned with equal weights ($w_1 = w_2 = w_3 = 1/3$) in Eq. (2) by default. Users can also choose different weights according to their needs.

6.2 Searching by a single paper

For a single-paper query, we selected 50 articles that were most similar to the query paper in each of the query paper's communities. Eq. (11) measures the similarity. The selected papers were the candidates for constructing evolution chains. To analyze the retrieval results more accurately, we chose two subfields that we were familiar with to do the experiments: hyperspectral imagery classification and hyperspectral imagery unmixing. Fig. 9 shows the PEG of the query paper "Spectral and spatial classification of hyperspectral data using SVMs and morphological profiles." This article is about the application of morphological profiles in hyperspectral imagery classification using SVM (support vector machine) as the classifier. It belongs to community 11 and community 14 with probabilities of 0.65 and 0.34, respectively; thus, the PEG is a combination of two chains of length six. As the word clouds show, community 11 contains the keyword "morphological," while community 14 includes articles relating to machine learning. Correspondingly, the first chain of the PEG in Fig. 9 focuses on classification of hyperspectral imagery with the features relating to morphological profiles. The second chain describes various SVM based classification methods.

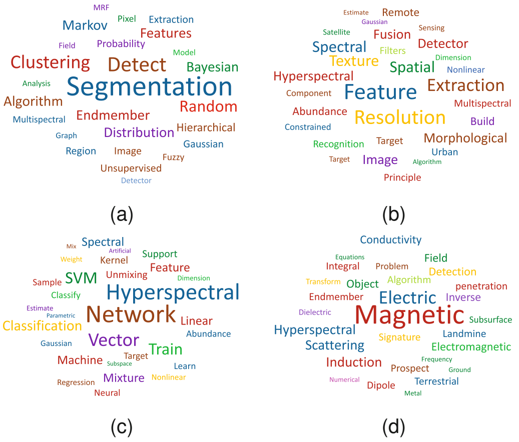


Fig. 8 Word clouds for various communities: (a) word cloud of community 9; (b) word cloud of community 11; (c) word cloud of community 14; (d) word cloud of community 30

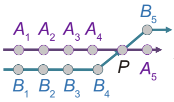
Fig. 10 displays the PEG of the query paper "Constrained nonnegative matrix factorization for hyperspectral unmixing." The paper belongs to community 9 and community 14 with probabilities of 0.36 and 0.35, respectively; thus, the PEG consists of two evolution chains. The length of the chain is set to five. Chain 1 focuses on using nonnegative matrix factorization to unmix hyperspectral imagery with different constraints. Chain 2 has two common papers with chain 1, but it focuses more on applying linear methods to do mixture analysis, which is a task substantially equivalent to the unmixing problem.

6.3 Searching by two papers

For two-paper retrieval, our system selected 50 papers that were most relevant to the query papers

Input: Spectral and spatial classification of hyperspectral data using SVMs and morphological profiles

Output:



Chain 1

A_1: Morphological transformations and feature extraction of urban data with high spectral and spatial resolution. (July 2003)

A_2: Decision level fusion in classification of hyperspectral data from urban areas. (Sept. 2004)

A_3: Classification of hyperspectral data from urban areas based on extended morphological profiles. (March 2005)

A_4: High-resolution multispectral image classification over urban areas by image segmentation and extended morphological profile. (July 2006)

P: Spectral and spatial classification of hyperspectral data using SVMs and morphological profiles. (Nov. 2008)

A_5: Classification of hyperspectral images with extended attribute profiles and feature extraction techniques. (July 2010)

Chain 2

B_1: Support vector machines for classification of hyperspectral remote-sensing images. (June 2002)

B_2: Source based feature extraction for support vector machines in hyperspectral classification. (Sept. 2004)

B_3: Transductive SVMs for semisupervised classification of hyperspectral data. (July 2005)

B_4: A combined support vector machines classification based on decision fusion. (July 2005)

P: Spectral and spatial classification of hyperspectral data using SVMs and morphological profiles. (Nov. 2008)

B_5: Spectral and spatial classification of hyperspectral data using SVMs and Gabor textures. (July 2011)

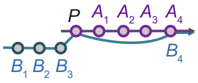
Fig. 9 Paper evolution graph of the query paper "Spectral and spatial classification of hyperspectral data using SVMs and morphological profiles"

using Eq. (11) in each of the communities shared by the query papers. Fig. 11 shows the PEG constructed for the query papers "Kernel-based methods for hyperspectral image classification" and "Kernel nonparametric weighted feature extraction for hyperspectral image classification." The two papers both focus on hyperspectral imagery classification using kernel methods. All papers in the chain apply kernel methods in hyperspectral imagery classification.

Fig. 12 shows the PEG constructed for the query papers "Endmember extraction from highly mixed data using minimum volume constrained nonnegative matrix factorization" and "A novel strategy of nonnegative-matrix-factorization-based polarimetric ship detection." Both papers address hyperspectral imagery detection using nonnegative matrix factorization. The former paper uses NMF to address a feature extraction task, which is a general detection problem, while the latter paper extends the method to solve a specific ship detection problem.

Input: Constrained nonnegative matrix factorization for hyperspectral unmixing

Output:



Chain 1

P: Constrained nonnegative matrix factorization for hyperspectral unmixing. (Jan. 2009)

A_1: Minimum dispersion constrained nonnegative matrix factorization to unmix hyperspectral data. (June 2010)

A_2: A novel approach for hyperspectral unmixing based on nonnegative matrix factorization. (July 2010)

A_3: Minimum endmember-wise distance constrained nonnegative matrix factorization for spectral mixture analysis of hyperspectral images. (July 2011)

A_4: Hyperspectral unmixing via $L_{1/2}$ sparsity-constrained nonnegative matrix factorization. (Nov. 2011)

Chain 2

B_1: Fully constrained least squares linear spectral mixture analysis method for material quantification in hyperspectral imagery. (March 2001)

B_2: Linear spectral random mixture analysis for hyperspectral imagery. (Feb. 2002)

B_3: Weighted abundance-constrained linear spectral mixture analysis. (Feb. 2006)

P: Constrained nonnegative matrix factorization for hyperspectral unmixing. (Jan. 2009)

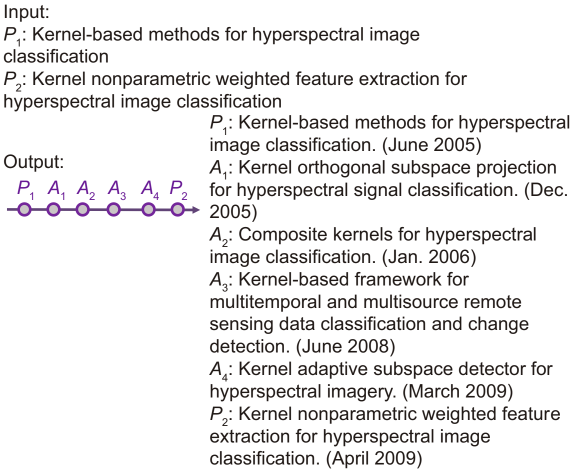
B_4: Hyperspectral unmixing via $L_{1/2}$ sparsity-constrained nonnegative matrix factorization. (Nov. 2011)

Fig. 10 Paper evolution graph of the query paper "Constrained nonnegative matrix factorization for hyperspectral unmixing"

6.4 Searching by keywords

For a keyword search, our system chose 100 papers that were most relevant to the keyword as candidates in constructing a PEG. Fig. 13 shows the PEG generated for query word "hyperspectral classification." The length of the evolution chain is set to six. Since the candidate papers are from community 11 and community 14, the PEG consists of two evolution chains extracted from the two communities. Both chains of papers solve the problem of hyperspectral imagery classification. Chain 1 follows the technique of classifying hyperspectral pixels using SVM, a classifier that is well known for its superior performance on small datasets with high-dimensional features. Chain 2 focuses on subspace method based classification methods, where the high dimensional data were first projected to a low-dimensional subspace before classification.

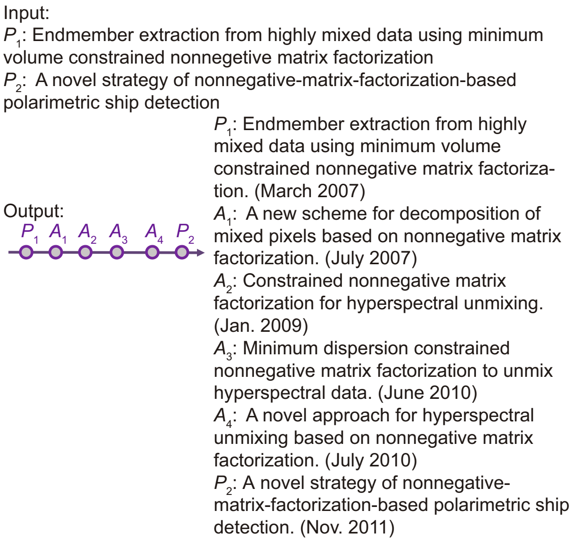
Fig. 14 shows the PEG generated for keyword "SAR denoising." The evolution chain consists of a set of papers on denoising synthetic-aperture radar (SAR) images in the wavelet and curvelet domains.


Input:
P_1: Kernel-based methods for hyperspectral image classification
P_2: Kernel nonparametric weighted feature extraction for hyperspectral image classification

Output:

P_1: Kernel-based methods for hyperspectral image classification. (June 2005)
A_1: Kernel orthogonal subspace projection for hyperspectral signal classification. (Dec. 2005)
A_2: Composite kernels for hyperspectral image classification. (Jan. 2006)
A_3: Kernel-based framework for multitemporal and multisource remote sensing data classification and change detection. (June 2008)
A_4: Kernel adaptive subspace detector for hyperspectral imagery. (March 2009)
P_2: Kernel nonparametric weighted feature extraction for hyperspectral image classification. (April 2009)

Fig. 11 Paper evolution graph of two-paper retrieval focusing on hyperspectral imagery classification


Input:
P_1: Endmember extraction from highly mixed data using minimum volume constrained nonnegetive matrix factorization
P_2: A novel strategy of nonnegative-matrix-factorization-based polarimetric ship detection

Output:

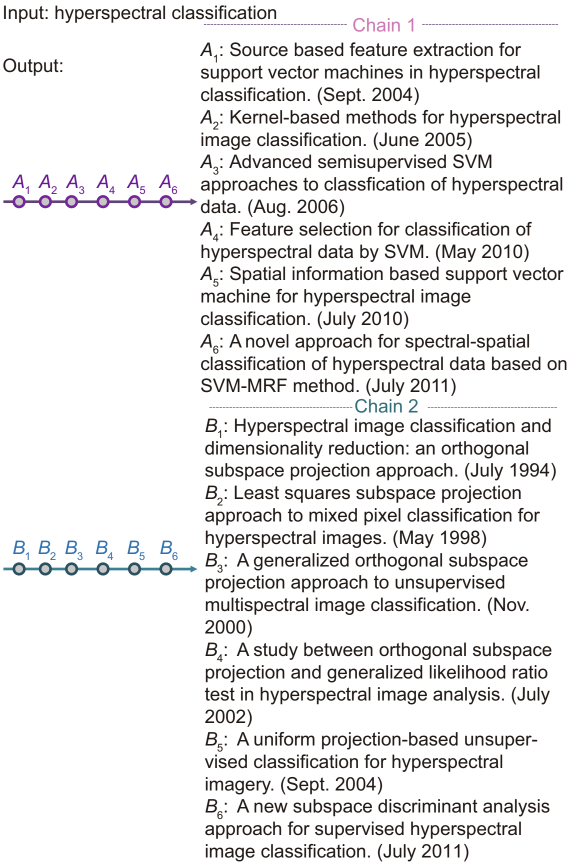
P_1: Endmember extraction from highly mixed data using minimum volume constrained nonnegative matrix factorization. (March 2007)
A_1: A new scheme for decomposition of mixed pixels based on nonnegative matrix factorization. (July 2007)
A_2: Constrained nonnegative matrix factorization for hyperspectral unmixing. (Jan. 2009)
A_3: Minimum dispersion constrained nonnegative matrix factorization to unmix hyperspectral data. (June 2010)
A_4: A novel approach for hyperspectral unmixing based on nonnegative matrix factorization. (July 2010)
P_2: A novel strategy of nonnegative-matrix-factorization-based polarimetric ship detection. (Nov. 2011)

Fig. 12 Paper evolution graph of the two-paper retrieval regarding unmixing and detection using nonnegative matrix factorization

6.5 Incorporating user preference

Our system can use the three relations between papers to generate PEGs from different views. We integrate user preferences into our framework. At the soft-clustering stage, users are allowed to choose different weights (w_1, w_2, w_3) for the three relations according to their needs. When no user preference is provided, we set $w_1 = w_2 = w_3 = 0.33$.

When a large weight is chosen on the "content" relation, the clustering approach generates communities based mainly on the content similarity of papers. As a result, papers in an extracted chain are likely to have a high similarity in content. Fig. 15a shows the PEG of the query paper, "Hyperspectral subspace identification", with $w_1 = 0.2$, $w_2 = 0.6$, and $w_3 = 0.2$. It shows that all papers in the chain focus on the subspace-based method in hyperspectral imagery.

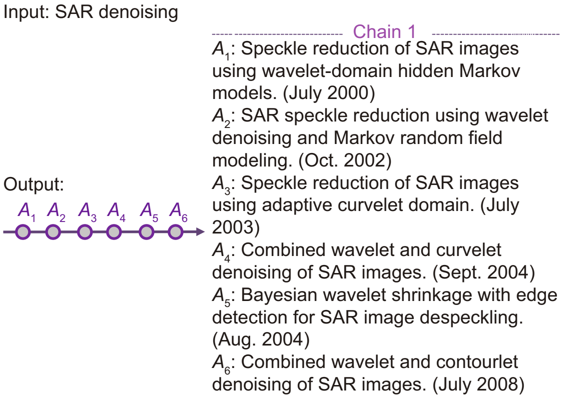

Input: hyperspectral classification

Output:

A_1: Source based feature extraction for support vector machines in hyperspectral classification. (Sept. 2004)
A_2: Kernel-based methods for hyperspectral image classification. (June 2005)
A_3: Advanced semisupervised SVM approaches to classfication of hyperspectral data. (Aug. 2006)
A_4: Feature selection for classification of hyperspectral data by SVM. (May 2010)
A_5: Spatial information based support vector machine for hyperspectral image classification. (July 2010)
A_6: A novel approach for spectral-spatial classification of hyperspectral data based on SVM-MRF method. (July 2011)

B_1: Hyperspectral image classification and dimensionality reduction: an orthogonal subspace projection approach. (July 1994)
B_2: Least squares subspace projection approach to mixed pixel classification for hyperspectral images. (May 1998)
B_3: A generalized orthogonal subspace projection approach to unsupervised multispectral image classification. (Nov. 2000)
B_4: A study between orthogonal subspace projection and generalized likelihood ratio test in hyperspectral image analysis. (July 2002)
B_5: A uniform projection-based unsupervised classification for hyperspectral imagery. (Sept. 2004)
B_6: A new subspace discriminant analysis approach for supervised hyperspectral image classification. (July 2011)

Fig. 13 Paper evolution graph for "hyperspectral classification" retrieval

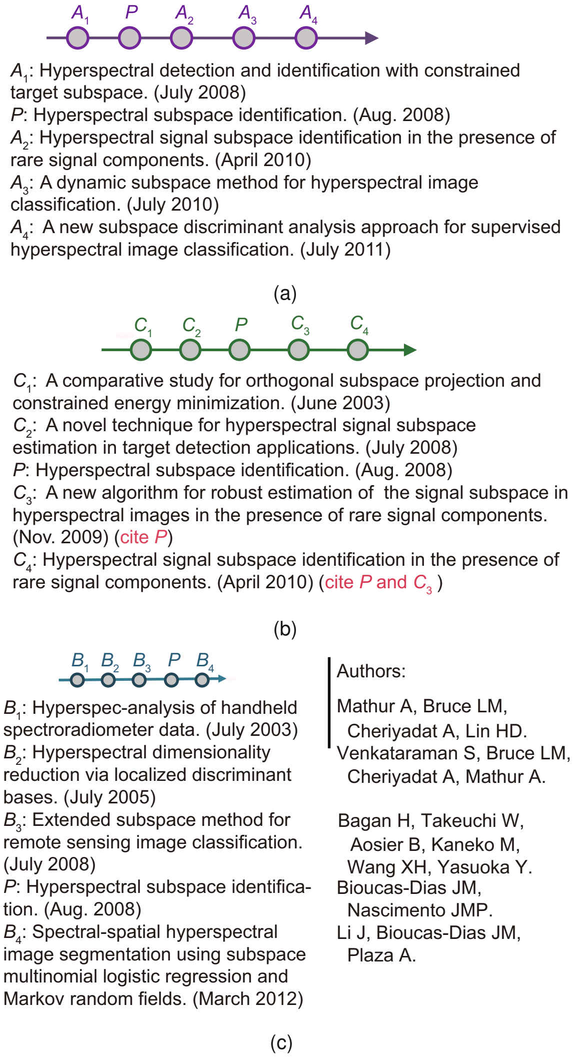

Input: SAR denoising

Output:

A_1: Speckle reduction of SAR images using wavelet-domain hidden Markov models. (July 2000)
A_2: SAR speckle reduction using wavelet denoising and Markov random field modeling. (Oct. 2002)
A_3: Speckle reduction of SAR images using adaptive curvelet domain. (July 2003)
A_4: Combined wavelet and curvelet denoising of SAR images. (Sept. 2004)
A_5: Bayesian wavelet shrinkage with edge detection for SAR image despeckling. (Aug. 2004)
A_6: Combined wavelet and contourlet denoising of SAR images. (July 2008)

Fig. 14 Paper evolution graph for "SAR denoising" retrieval

The citation relation reveals the correlation between papers if they do not have a high similarity in content. If users are interested in discovering papers that have a citation relationship, he/she can choose



A_1: Hyperspectral detection and identification with constrained target subspace. (July 2008)
P: Hyperspectral subspace identification. (Aug. 2008)
A_2: Hyperspectral signal subspace identification in the presence of rare signal components. (April 2010)
A_3: A dynamic subspace method for hyperspectral image classification. (July 2010)
A_4: A new subspace discriminant analysis approach for supervised hyperspectral image classification. (July 2011)

(a)

C_1: A comparative study for orthogonal subspace projection and constrained energy minimization. (June 2003)
C_2: A novel technique for hyperspectral signal subspace estimation in target detection applications. (July 2008)
P: Hyperspectral subspace identification. (Aug. 2008)
C_3: A new algorithm for robust estimation of the signal subspace in hyperspectral images in the presence of rare signal components. (Nov. 2009) (cite P)
C_4: Hyperspectral signal subspace identification in the presence of rare signal components. (April 2010) (cite P and C_3)

(b)

B_1: Hyperspec-analysis of handheld spectroradiometer data. (July 2003)
B_2: Hyperspectral dimensionality reduction via localized discriminant bases. (July 2005)
B_3: Extended subspace method for remote sensing image classification. (July 2008)
P: Hyperspectral subspace identification. (Aug. 2008)
B_4: Spectral-spatial hyperspectral image segmentation using subspace multinomial logistic regression and Markov random fields. (March 2012)

Authors:
Mathur A, Bruce LM, Cheriyadat A, Lin HD.
Venkataraman S, Bruce LM, Cheriyadat A, Mathur A.
Bagan H, Takeuchi W, Aosier B, Kaneko M, Wang XH, Yasuoka Y.
Bioucas-Dias JM, Nascimento JMP.
Li J, Bioucas-Dias JM, Plaza A.

(c)

Fig. 15 Paper evolution graph emphasizing content coherence (a), citation coherence (b), and author coherence (c)

a large weight on the "citation" relation. In this case, papers having citation relationships are likely to be clustered into the same community. As a result, papers in an evolution chain not only share the same topic but also have citation relation along the chain. Fig. 15b shows the PEG of the query paper, "Hyperspectral subspace identification", with $w_1 = 0.6$, $w_2 = 0.2$, and $w_3 = 0.2$. In the evolution chain, both C_3 and C_4 cite P, and C_4 cites C_3.

In the case of a large weight for the "author" relation, our clustering approach is inclined to generate clusters based on the authorship information; i.e., papers published by the same authors tend to be clustered into communities. As a result, chains extracted from such communities are likely to consist of papers sharing the same authors. Fig. 15c shows the PEG for the query paper "Hyperspectral subspace identification" with $w_1 = 0.2$, $w_2 = 0.2$, and $w_3 = 0.6$, where the first two papers share two same authors and the last two papers share a common author.

7 Evaluation

A common method to evaluate the performance of information retrieval (IR) systems is to test the algorithms on labeled datasets and calculate some standard metrics such as retrieval precision. However, these datasets are designed for list-output models and thus they are not suitable for evaluating our structural retrieval system. As a result, we evaluated our algorithm on real-world datasets and tested three metrics: (1) accuracy (showing how strong the articles retrieved in a PEG are relevant to the query); (2) coherence (showing the topic coherence of an evolution chain—specifically, we measured the coherence by the definition given in Eq. (10) and also invited some domain experts to evaluate the topic coherence of the results); (3) helpfulness (we conducted a user study to see how a PEG can help beginners understand a new academic article as well as obtain the big picture on an unfamiliar research domain).

Because the proposed method returns structural retrieval results, it cannot be compared directly with most of the other retrieval systems that return isolated results. In this study, we designed a way to compare our system with Google Scholar, IEEE Xplore, and Web of Science by manually constructing evolution chains from the retrieval results of the compared systems.

7.1 Accuracy

In this study, we evaluated the accuracy of the PEG, i.e., whether the articles in the PEG are relevant to the query. To this end, we compared the retrieval results of our system with the results of Google Scholar, IEEE Xplore, and Web of Science to see how our retrieval results are ranked in other systems. Our assumption is that if the articles in the PEG are at the top of the returned list in the other systems, then it indicates that the accuracy of the PEG is high. We created two tasks corresponding to two keyword queries "SAR denoising" and "Hyperspectral classification." For a fair comparison, the search region in the compared systems was set to TGRS and IGARSS. Fig. 16a shows the comparison of results of the query "Hyperspectral classification." Fig. 16b shows the comparison of results of the query "SAR denoising." The data in the compared systems were collected on March 14, 2015. The green line in Fig. 16a indicates that 60% of the papers in the PEG are ranked before 50 in Google Scholar. The comparison shows that a large proportion of papers in the PEG ranked on top of the retrieval list of the other systems. Note that since our goal is to find those articles that are most topically coherent, finding the most influential/relevant papers is not the main desired property of PEG.

7.2 Coherence

In this subsection, our goal is to evaluate the topic coherence of the PEG. This is done by measuring the topic coherence defined in Eq. (10). We also invited several domain experts to manually score the coherence of the retrieval results. We compared our method with the retrieval results from Google Scholar, IEEE Xplore, and Web of Science.

Since the retrieval results of the compared systems are isolated, they cannot be compared with the PEG directly. For a fair comparison, we manually constructed evolution chains from the retrieved papers of the compared systems. Again, the papers in the compared systems were restricted to publication in TGRS and IGARSS. The comparison was done under three types of query. The length of the evolution chain was set to six in all experiments. For different types of queries, we constructed chains from the compared systems as follows:

For a single-paper query, we first selected the top five papers in the returned list from the compared systems. Then the selected papers together with the query paper were arranged in chronological order to form an evolution chain of length six. When the PEG contained more than one chain, five more papers were selected from the compared systems to form each additional chain.

For a two-paper query (p_s or p_t), we constructed a chain from the systems in two steps. First, for each of the two queries, we selected its two most relevant papers whose publication time was between that of p_s and p_t. Then the four selected results together with p_s and p_t were arranged in chronological order to form an evolution chain.

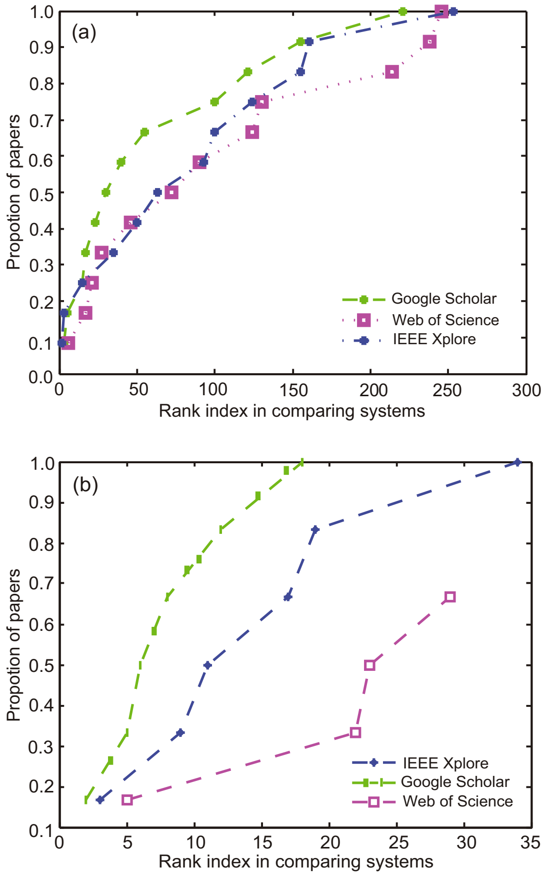


Fig. 16 Proportion of papers in the paper evolution graph covered by different systems: (a) comparison for "hyperspectral classification;" (b) comparison for "hyperspectral unmixing." References to color refer to the online version of this figure

For a keyword query, we first selected the top six papers from the compared systems to form an evolution chain. Six more papers were selected for each additional chain in the PEG. Then the papers were arranged in chronological order to form the same number of chains as in the PEG.

We created 10 tasks for each type of query and calculated the topic coherence as defined in Eq. (10). Fig. 17 shows the average coherence of the tasks for different systems. The results demonstrate that the evolution chains in the PEG are much more topically cohesive than the other systems' results.

We also evaluated the coherence by asking a group of domain experts to score the chains generated by different systems. Since the PEG is unique in structure (chains intersect with each other), we were unable to do a double-blind comparison study in its original form. To deal with this problem, we separated the combined chains in the PEG into independent chains so that the experts could not differentiate between the different systems. The expert group was composed of six experts, including four teachers and two PhD students. Each expert had at least two years of research experience in the remote sensing domain. We created five tasks for each type of query in the study. The experts were asked to score the correlation between adjacent papers with 0–5 points (0 means not related at all). Since a high correlation between the adjacent papers does not guarantee a coherent topic along the whole chain, we also asked the experts to grade the topic coherence of the overall chain with 0–5 points. One of the grading tables is shown in Fig. 18.

The average grading results for all tasks are given in Table 2. In the study, the experts gave very high marks to the retrieval results presented by PEG. The results show that the correlations between adjacent papers in PEG are much stronger than those in other systems. In addition, the topic along the overall chain of PEG was much more focused than in the compared systems.

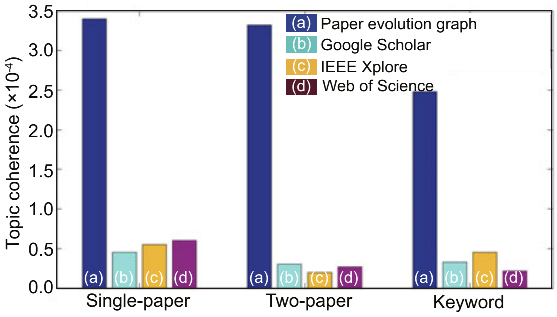
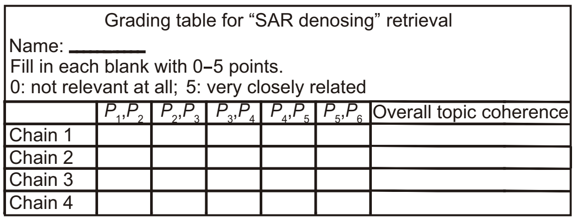


Fig. 17 Comparison of topic coherence



Grading table for “SAR denosing” retrieval

Name: ______

Fill in each blank with 0–5 points.

0: not relevant at all; 5: very closely related

	P_1,P_2	P_2,P_3	P_3,P_4	P_4,P_5	P_5,P_6	Overall topic coherence
Chain 1						
Chain 2						
Chain 3						
Chain 4						

Fig. 18 Grading table for “SAR denoising” retrieval

Table 2 Average coherence score of the expert group

System	Coherence	
	Adjacent score	Overall score
Paper evolution graph	4.10	4.53
Google Scholar	2.40	2.45
Web of Science	1.64	2.20
IEEE Xplore	1.35	1.78

7.3 Helpfulness in digesting new information

In this subsection, we analyze whether PEG can help beginners digest new information when they are facing an unfamiliar research domain. To achieve this goal, we designed a user study to see how PEG can help users understand a new paper as well as comprehend the big picture of a new research domain. We recruited 16 students in our college to do the study. All of the participants met two conditions: (1) they were able to read academic articles in the remote sensing domain; (2) they did not know the domain well in advance. In this study, we designed three tasks for a single-paper query and a keyword query, respectively. In the single-paper query, we aimed to find out how PEG can help beginners answer specific questions related to the query paper. In the keyword search tasks, the users were asked to describe some important concepts and answer specific questions regarding the query domain. The queries and questions were designed specifically for beginners by a PhD student who had rich research experience in the remote sensing domain. Again, we compared the level of knowledge attained by beginners using our prototype versus Google Scholar, IEEE Xplore, and Web of Science. The approach used to select papers from the compared systems is described in Section 7.2. The students were divided into four groups to score the results from the different systems.

At the first stage of the single-paper query test, each participant was asked to finish a questionnaire with questions regarding the query paper. One correct answer added one point. The scores were recorded to measure the students’ pre-knowledge about the paper. After they finished the questionnaires, students in different groups were asked to read the query paper with the help of a set of retrieved papers from the compared retrieval systems. They were allowed to modify the questionnaires while reading. After they finished their final questionnaires, the improved scores of each student were calculated. We believe that the higher the improved scores were, the more useful the retrieved papers were in helping the beginners digest a paper.

The second user study was to measure the helpfulness of PEG in aiding users to grasp the general information in unfamiliar domains. Again, before reading any article the students were asked to finish a questionnaire regarding the three research domains. Then one group of students was presented with PEG results generated by keyword search, while students in the other groups were presented with papers retrieved by the compared systems. They were allowed to modify their questionnaires while reading.

We recorded the final scores and calculated the improved scores for all of the six tasks. The average improved scores of the four groups are given in Table 3. The results demonstrated that PEG performs better in helping beginners digest the details of a paper and comprehend the general knowledge within a research domain.

Table 3 Improved scores of different groups

System	Improved score	
	single-paper	keywords
Paper evolution graph	4.83	3.20
Google Scholar	3.15	2.04
Web of Science	2.58	1.28
IEEE Xplore	2.11	1.67

8 Conclusions and future work

In this paper, we have presented a method for creating structured paper retrieval results, which we call paper evolution graph (PEG). The PEG explicitly shows the multi-view relationships between the retrieved papers by a combination of a set of evolution chains. Each chain consists of a sequence of topically cohesive papers; different chains follow different topics of the query. Three types of information (content, author, and citation) are used in our system, to which users were allowed to attribute different weights to generate an evolution graph emphasizing different aspects. Our system supports keyword search, single-paper search, and two-paper search to meet different user requirements.

In the future, we plan to enlarge our dataset and invite more researchers to use and evaluate our system. In addition, we intend to use more types of information such as a paper's academic influence. Another important issue is to extend our system to time-varying datasets.

References

Agrawal R, Gollapudi S, Halverson A, et al., 2009. Diversifying search results. Proc 2nd ACM Int Conf on Web Search and Data Mining, p.5-14. https://doi.org/10.1145/1498759.1498766

Ahmed A, Ho Q, Eisenstein J, et al., 2011. Unified analysis of streaming news. Proc 20th Int Conf on World Wide Web, p.267-276. https://doi.org/10.1145/1963405.1963445

Aljaber B, Stokes N, Bailey J, et al., 2010. Document clustering of scientific texts using citation contexts. *Inform Retriev*, 13(2):101-131. https://doi.org/10.1007/s10791-009-9108-x

Allan J, Gupta R, Khandelwal V, 2001. Temporal summaries of new topics. Proc 24th Annual Int ACM SIGIR Conf on Research and Development in Information Retrieval, p.10-18. https://doi.org/10.1145/383952.383954

Bader BW, Kolda TG, 2006. Algorithm 862: Matlab tensor classes for fast algorithm prototyping. *ACM Trans Math Softw*, 32(4):635-653. https://doi.org/10.1145/1186785.1186794

Banerjee A, Basu S, Merugu S, 2007. Multi-way clustering on relation graphs. Proc SIAM Int Conf on Data Mining, p.145-156. https://doi.org/10.1137/1.9781611972771.14

Blei DM, Ng AY, Jordan MI, 2003. Latent Dirichlet allocation. *J Mach Learn Res*, 3:993-1022.

Bolelli L, Ertekin Ş, Giles CL, 2009. Topic and trend detection in text collections using latent Dirichlet allocation. European Conf on Information Retrieval, p.776-780. https://doi.org/10.1007/978-3-642-00958-7_84

Brin S, Page L, 1998. The anatomy of a large-scale hypertextual Web search engine. *Comput Netw ISDN Syst*, 30(1-7):107-117. https://doi.org/10.1016/S0169-7552(98)00110-X

Butler D, 2004. Science searches shift up a gear as Google starts scholar engine. *Nature*, 432(7016):423. https://doi.org/10.1038/432423a

Campbell I, 2000. Interactive evaluation of the ostensive model using a new test collection of images with multiple relevance assessments. *Inform Retriev*, 2(1):89-114. https://doi.org/10.1023/A:1009902203782

Chen H, Karger DR, 2006. Less is more: probabilistic models for retrieving fewer relevant documents. Proc 29th Annual Int ACM SIGIR Conf on Research and Development in Information Retrieval, p.429-436. https://doi.org/10.1145/1148170.1148245

Chen P, Xie H, Maslov S, et al., 2007. Finding scientific gems with Google's PageRank algorithm. *J Inform*, 1(1):8-15. https://doi.org/10.1016/j.joi.2006.06.001

Garfield E, 1979. Citation Indexing: Its Theory and Application in Science, Technology, and Humanities. Wiley, New York, USA.

Gohr A, Hinneburg A, Schult R, et al., 2009. Topic evolution in a stream of documents. Proc SIAM Int Conf on Data Mining, p.859-872. https://doi.org/10.1137/1.9781611972795.74

He Q, Chen B, Pei J, et al., 2009. Detecting topic evolution in scientific literature: how can citations help? Proc 18th ACM Conf on Information and Knowledge Management, p.957-966. https://doi.org/10.1145/1645953.1646076

Jo Y, Hopcroft JE, Lagoze C, 2011. The web of topics: discovering the topology of topic evolution in a corpus. Proc 20th Int Conf on World Wide Web, p.257-266. https://doi.org/10.1145/1963405.1963444

Kempe D, Kleinberg J, Tardos É, 2003. Maximizing the spread of influence through a social network. Proc 9th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining, p.137-146. https://doi.org/10.1145/956750.956769

Kleinberg J, 1999. Authoritative sources in a hyperlinked environment. *J ACM*, 46(5):604-632. https://doi.org/10.1145/324133.324140

Kleinberg J, 2003. Bursty and hierarchical structure in streams. *Data Min Knowl Discov*, 7(4):373-397. https://doi.org/10.1023/A:1024940629314

Lafferty J, Zhai CX, 2001. Document language models, query models, and risk minimization for information retrieval. Proc 24th Annual Int ACM SIGIR Conf on Research and Development in Information Retrieval, p.111-119. https://doi.org/10.1145/383952.383970

Lavrenko V, Croft WB, 2001. Relevance based language models. Proc 24th Annual Int ACM SIGIR Conf on Research and Development in Information Retrieval, p.120-127. https://doi.org/10.1145/383952.383972

Lin YR, Sun JM, Castro P, et al., 2009. MetaFac: community discovery via relational hypergraph factorization. Proc 15th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining, p.527-536. https://doi.org/10.1145/1557019.1557080

Long B, Zhang ZF, Wu XY, et al., 2006. Spectral clustering for multi-type relational data. Proc 23rd Int Conf on Machine Learning, p.585-592. https://doi.org/10.1145/1143844.1143918

Makkonen J, 2003. Investigations on event evolution in TDT. Proc Conf of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: Proc HLT-NAACL Student, p.43-48. https://doi.org/10.3115/1073416.1073424

Mei QZ, Zhai CX, 2005. Discovering evolutionary theme patterns from text: an exploration of temporal text mining. Proc 11th ACM SIGKDD Int Conf on Knowledge Discovery in Data Mining, p.198-207. https://doi.org/10.1145/1081870.1081895

Mei QZ, Liu C, Su H, et al., 2006. A probabilistic approach to spatiotemporal theme pattern mining on weblogs. Proc 15th Int Conf on World Wide Web, p.533-542. https://doi.org/10.1145/1135777.1135857

Nallapati R, Feng A, Peng FC, et al., 2004. Event threading within news topics. Proc 13th ACM Int Conf on Information and Knowledge Management, p.446-453. https://doi.org/10.1145/1031171.1031258

Narin F, 1976. Evaluative Bibliometrics: the Use of Publication and Citation Analysis in the Evaluation of Scientific Activity. Computer Horizons, Inc., Washington, DC, USA.

Newman ME, 2001. Scientific collaboration networks. I. network construction and fundamental results. *Phys Rev E*, 64:016131. https://doi.org/10.1103/PhysRevE.64.016131

Robertson SE, 1977. The probability ranking principle in IR. *J Doc*, 33(4):294-304. https://doi.org/10.1108/eb026647

Rosen-Zvi M, Griffiths T, Steyvers M, et al., 2004. The author-topic model for authors and documents. Proc 20th Conf on Uncertainty in Artificial Intelligence, p.487-494.

Salton G, 1971. The Smart Retrieval System—Experiments in Automatic Document Processing. Prentice-Hall, Englewood Cliffs, USA.

Schult R, Spiliopoulou M, 2006. Discovering emerging topics in unlabelled text collections. Proc 10th East European Conf on Advances in Databases and Information Systems, p.353-366. https://doi.org/10.1007/11827252_27

Shahaf D, Guestrin C, 2010. Connecting the dots between news articles. Proc 16th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining, p.623-632. https://doi.org/10.1145/1835804.1835884

Shahaf D, Guestrin C, Horvitz E, 2012. Trains of thought: generating information maps. Proc 21st Int Conf on World Wide Web, p.899-908. https://doi.org/10.1145/2187836.2187957

Shen XH, Zhai CX, 2005. Active feedback in ad-hoc information retrieval. Proc 28th Annual Int ACM SIGIR Conf on Research and Development in Information Retrieval, p.59-66. https://doi.org/10.1145/1076034.1076047

Small H, 1973. Co-citation in the scientific literature: a new measure of the relationship between two documents. *J Am Soc Inform Sci*, 24(4):265-269. https://doi.org/10.1002/asi.4630240406

Spiliopoulou M, Ntoutsi I, Theodoridis Y, et al., 2006. MONIC: modeling and monitoring cluster transitions. Proc 12th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining, p.706-711. https://doi.org/10.1145/1150402.1150491

Steyvers M, Smyth P, Rosen-Zvi M, et al., 2004. Probabilistic author-topic models for information discovery. Proc 10th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining, p.306-315. https://doi.org/10.1145/1014052.1014087

Tang J, Zhang J, Yao LM, et al., 2008. ArnetMiner: extraction and mining of academic social networks. Proc 14th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining, p.990-998. https://doi.org/10.1145/1401890.1402008

Yan R, Wan XJ, Otterbacher J, et al., 2011. Evolutionary timeline summarization: a balanced optimization framework via iterative substitution. Proc 34th Int ACM SIGIR Conf on Research and Development in Information Retrieval, p.745-754. https://doi.org/10.1145/2009916.2010016

Yu J, Mohan S, Putthividhya D, et al., 2014. Latent Dirichlet allocation based diversified retrieval for e-commerce search. Proc 7th ACM Int Conf on Web Search and Data Mining, p.463-472. https://doi.org/10.1145/2556195.2556215

Yu L, Liu C, Zhang ZK, 2015. Multi-linear interactive matrix factorization. *Knowl Based Syst*, 85:307-315. https://doi.org/10.1016/j.knosys.2015.05.016

Zhou D, Ji X, Zha HY, et al., 2006. Topic evolution and social interactions: how authors effect research. Proc 15th ACM Int Conf on Information and Knowledge Management, p.248-257. https://doi.org/10.1145/1183614.1183653

Zhu SH, Yu K, Chi Y, et al., 2007. Combining content and link for classification using matrix factorization. Proc 30th Annual Int ACM SIGIR Conf on Research and Development in Information Retrieval, p.487-494. https://doi.org/10.1145/1277741.1277825