Learning Multiple Graphs for Document Recommendations

Ding Zhou[*]
Facebook Inc.
156 University Avenue
Palo Alto, CA 94301

Shenghuo Zhu   Kai Yu
NEC Labs America
10080 N Wolfe Road,
Cupertino, CA 95014

Xiaodan Song
Google Inc.
1600 Amphitheatre Pkway,
Mountain View, CA 94043

Belle L. Tseng
Yahoo! Inc.
701 First Avenue
Sunnyvale, CA 94089

Hongyuan Zha
College of Computing
Georgia Institute of
Technology
Atlanta, GA 30332

C. Lee Giles
Information Sciences and
Technology
Computer Science &
Engineering
Pennsylvania State University
University Park, PA 16802

ABSTRACT

The Web offers rich relational data with different semantics. In this paper, we address the problem of document recommendation in a digital library, where the documents in question are networked by citations and are associated with other entities by various relations. Due to the sparsity of a single graph and noise in graph construction, we propose a new method for combining multiple graphs to measure document similarities, where different factorization strategies are used based on the nature of different graphs. In particular, the new method seeks a single low-dimensional embedding of documents that captures their relative similarities in a latent space. Based on the obtained embedding, a new recommendation framework is developed using semi-supervised learning on graphs. In addition, we address the scalability issue and propose an incremental algorithm. The new incremental method significantly improves the efficiency by calculating the embedding for new incoming documents only. The new batch and incremental methods are evaluated on two real world datasets prepared from CiteSeer. Experiments demonstrate significant quality improvement for our batch method and significant efficiency improvement with tolerable quality loss for our incremental method.

General Terms

Algorithm, Experimentation

Keywords

Recommender Systems, Collaborative Filtering, Semi-supervised Learning, Social Network Analysis, Spectral Clustering

1. INTRODUCTION

Recommender systems continue to play important and new roles in business on the World Wide Web [11, 12, 14, 10, 13]. Per definition, the *recommender system* is an information filtering technique that seeks to identify a set of items that are likely of interest to users.

The most popular method adopted by contemporary recommender systems is *Collaborative Filtering* (CF), where the core assumption is that *similar* users on *similar* items express *similar* interests. The heart of memory-based CF methods is the measurement of similarity: either the similarity of users (a.k.a user-based CF) or the similarity of items (a.k.a items-based CF) or a hybrid of both. The user-based CF computes the similarity among users, usually based on user profiles or past behavior [14, 10], and seeks consistency in the predictions among similar users. But it is known that user-based CF often suffers from the *data sparsity problem* because most of the user-item ratings are missing in practice. The item-based CF, on the other hand, allows input of additional item-wise information and is also capable of capturing the interactions among them [11, 12]. This is a major advantage of item-based CF when it comes to dealing with items that are networked, which are usually encountered on the Web. For example, consider the problem of document recommendation in a digital library such as the CiteSeer (http://citeseer.ist.psu.edu). As illustrated in Fig. 1, let documents be denoted as vertices on a directed graph where the edges indicate their citations. The similarity among documents can be measured by their cocitations (cociting the same documents or being cocited by others) [1]. In this case, document B and C are similar because they are cocited by E.

Working with networked items for CF is of recent interest. Recent work approaches this problem by leveraging the item similarities measured on an item graph [12], modeling item similarities by an undirected graph and, given several vertices labeled interesting, perform label propagation to rank the remaining vertices. The key issue in label propagation on graphs is the measurement of vertex similarity, where related work simply borrows the recent results of the Laplacian on directed graphs [2] and semi-supervised learning of

[*] This work was done at The Pennsylvania State University.

[1] In fact, the term *cocitation* in this paper refers to two concepts in information sciences: *bibliographic coupling* and *cocitation.*

Copyright is held by the International World Wide Web Conference Committee (IW3C2). Distribution of these papers is limited to classroom use, and personal use by others.
WWW 2008, Beijing, China
ACM 978-1-60558-085-2/08/04.

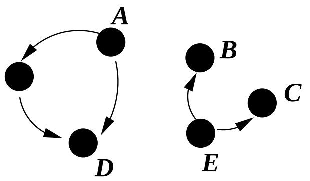
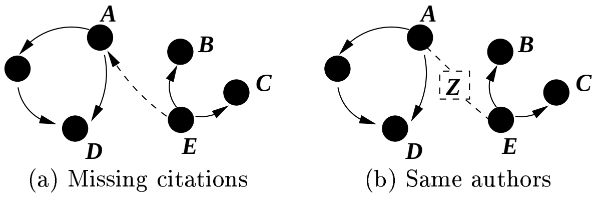


Figure 1: An example of citation graph.

graphs [18]. Nevertheless, using a single graph Laplacian to measure the item similarity can overfit in practice, especially for data on the Web, where the graphs tend to be noisy and sparse in nature. For example, if we revisit Fig. 1 and consider two quite common scenarios, as illustrated in Fig. 2, it is easy to see why measuring item similarities based on a single graph can sometimes cause problems. The first case is called *missing citations*, where for some reason a citation is missing (or equivalently is added) from the citation graph. Then the similarity between A and B (or C) will not be encoded in the graph Laplacian. The second case, called *same authors*, shows that if A and E are authored by the same researcher Z, using the citation graph only will not capture the similarity between D and B, which presumably should be similar because they are both cited by the author Z.



(a) Missing citations (b) Same authors

Figure 2: Two common scenarios: *missing citations* and *same authors*, which give rise to the problems for measuring item similarities based on a single citation graph.

Needless to say, the cases presented above are just two of the many problems caused by the *noise* and *sparsity* of the citation graph. Noise in a citation graph is a result of a missing citation link or an incorrect one. Fortunately, real world data can usually be described by different semantics or can be associated with other data. In the focus of this paper, where only relational data is concerned, we work with several graphs regarding the same set of items. For example, in the case of document recommendation, and in addition to the document citation graph, we also have a document-author bipartite graph that encodes the authorship, and a document-venue bipartite graph that indicates where the documents were published. Such relationship between documents and other objects can be used to improve the measurement of document similarity. The idea of this work is to combine multiple graphs to calculate the similarities among items. The items can be the full vertex set of a graph (as in the citation graph) or can be a subset of a graph (as in document-author bipartite graph) [2]. By doing so, we let data from different semantics regarding the same item set complement each other.

In this paper, we implement a model of learning from multiple graphs by seeking a single low-dimensional embedding of items that captures the relative similarities among them. Based on the obtained item embedding, we perform label propagation, giving rise to a new recommendation framework using semi-supervised learning on graphs. In addition, we address the scalability issue and propose an incremental version of our new method, where an approximate embedding is calculated only for the new items. The new methods are evaluated on two real world datasets prepared from CiteSeer. We compare the new batch method with a baseline modified from a recent semi-supervised learning algorithm on a directed graph and a basic user-based CF method using Singular Value Decomposition (SVD). Also, we compare the new incremental method with the new batch method in terms of recommendation quality and efficiency. We observe significant quality improvement in our batch method and significant efficiency improvement with tolerable quality loss for our incremental method.

The contributions of this work are: (1) We overcome the deficiency of a single graph (e.g. noise, sparsity) by combining multiple information sources (or graphs) via a joint factorization to learn rich yet compact representation of the items in question; (2) To ensure effectiveness and efficiency, we propose several novel factorization strategies tailored to the unique characteristics of each graph type, each becoming a sub-problem in the joint framework; (3) To handle the ever-growing volume of documents, we further develop an incremental updating algorithm that greatly improves the scalability, which is validated on two large real-world datasets.

The rest of this paper is organized as follows: Section 2 introduces how to realize recommendations using label propagation; Section 3 describes our method for learning item embedding from three general types of graphs; Section 4 further introduces the incremental version of our algorithm; Experiments are presented in Section 5; Section 6 discusses the related work; Conclusions are drawn in Section 7.

2. RECOMMENDATION BY LABEL PROPAGATION

Label propagation is one typical kind of *transductive learning* in the semi-supervised learning category where the goal is to estimate the labels of unlabeled data using other partially labeled data and their similarities. Label propagation on a network has many different applications. For example, recent work shows that trust between individuals can be propagated on social networks [7] and user interests can be propagated on item graphs for recommendations [12].

In this work, we focus on using label propagation for document recommendation in digital libraries. Let the document set be $\mathcal{D}$, where $|\mathcal{D}|$ is the number of documents. Suppose we are given the document citation graph $G_D = (V_D, E_D)$, which is an unweighted directed graph. Suppose the pairwise similarities among the documents are described by the matrix $S \in \mathbb{R}^{|\mathcal{D}|\times|\mathcal{D}|}$ measured based on G_D. A few documents have been labeled “interesting” while the remaining are not, denoted by positive and zero values in the label vector y. The goal is to find the score vector $f \in \mathbb{R}^{|\mathcal{D}|}$ where each element corresponds to the propagated interests. Then document recommendation can be performed by ranking the

[2] Note the difference between this work and the related work [16] where multiple graphs with the *same set of vertices* are combined.

documents by their interest scores. A recent approach addressed the graph label propagation problem by minimizing the regularization loss below [18]:

$$\Omega(f) \equiv f^T(I - S)f + \mu\|f - y\|^2, \tag{1}$$

where $\mu > 0$ is the regularization parameter. The first term is the cost function for the *smoothness constraint*, which prefers small differences in labels between nearby points; the second term is the *fitting constraint* that measures the difference of f from given data label y. Setting the $\partial\Omega(y)/\partial f = 0$, we can see that the solution f^* is essentially the solution to the linear equation:

$$(I - \alpha S)f^* = (1 - \alpha)y, \tag{2}$$

where $\alpha = 1/(1 + \mu)$. One solution to the above is given in a related work using a power method [18]:

$$f^{t+1} \leftarrow \alpha S f^t + (1 - \alpha)y \tag{3}$$

where f^0 is the random guess and $f^* = f^\infty$ is the solution. Here, notice that $\mathcal{L} = (I - \alpha S)$ is essentially a variant Laplacian on this graph using S as the adjacency matrix; and $\mathcal{K} = (I - \alpha S)^{-1} = \mathcal{L}^{-1}$ is the graph diffusion kernel. Thus, one essentially applies $f^* = (1 - \alpha)\mathcal{L}^{-1}y$ (or $f^* = (1 - \alpha)\mathcal{K}y$)to rank documents for recommendation.

Now the interesting question is how to calculate S (or equivalently the kernel $\mathcal{K}$) among the set $\mathcal{D}$. However, there has been limited amount of work on obtaining S. For graph data, recent work borrows the results from spectral graph theory [1, 2], where the similarity measures on both undirected and directed graphs have been given. For undirected graph, S_u is simply the normalized adjacency matrix:

$$S_u = \Pi^{-1/2} W \Pi^{-1/2} \tag{4}$$

where Π is a diagonal matrix such that $We = \Pi e$ and e is an all-one column vector. For directed graph, where the adjacency matrix is first normalized as a random walk transition matrix P $(= \Pi^{-1}W)$, the similarity measure S_d is calculated as:

$$S_d = \frac{\Phi^{1/2} P \Phi^{-1/2} + \Phi^{-1/2} P^T \Phi^{1/2}}{2} \tag{5}$$

where Φ is a diagonal matrix where each diagonal contains the stationary probability on the corresponding vertex [3].

Note that the similarity measures given above are derived from a single graph on $\mathcal{D}$. However, many real world data can be described by multiple graphs, including those within $\mathcal{D}$ and between $\mathcal{D}$ and another set. Such information is of more importance to combine especially when the a single view of the data is sparse or even incomplete. In the following, we introduce a new way to integrate three general types of graphs. Instead of estimating S directly, we seek to learn a low-dimensional latent linear space.

3. LEARNING MULTIPLE GRAPHS

The immediate goal of this section is to determine the relative positions of all documents in a k-dimensional latent semantic space, say $X \in \mathbb{R}^{|\mathcal{D}| \times k}$, which will combine the social inferences in document citations, authorship and venues. In the sequel, we assume k is a prescribed parameter which we do not seek to determine automatically. Note a contribution of this work is the different strategies used for different graphs based on their characteristics, which are described in the following subsections.

We begin by a formulation of our problem. Let $\mathcal{D}$, $\mathcal{A}$, $\mathcal{V}$ be the sets of documents, authors and venues and $|\mathcal{D}|$, $|\mathcal{A}|$, $|\mathcal{V}|$ be their sizes. We have three graphs, one directed graph G_D on $\mathcal{D}$; one bipartite graph G_{DA} between $\mathcal{D}$ and $\mathcal{A}$; and one bipartite graph G_{DV} between $\mathcal{D}$ and $\mathcal{V}$, which describe the relationship among documents, between documents and authors, and between documents and venues. Let the adjacency matrices of G_D, G_{DA}, G_{DV} be D, A and V. We assume all relationships in question are described by non-negative values. For example, G_D can be considered as to describe the citation relationship among $\mathcal{D}$ and $D_{i,j} = 1$ if document d_i cites d_j ($D_{i,j} = 0$ if otherwise); G_A can be considered as the authorship relationship (an author composes a document) or the citation relationship (an author cites a document) between $\mathcal{D}$ and $\mathcal{A}$.

3.1 Learning from Citation Matrix: D

In this section, we relate the document embedding X to the citation matrix D, which is the adjacency matrix of the the directed graph G_D.

The citation matrix D include two kinds of document co-occurrences: cociting and being cocited. A cociting relationship among a set of documents means that they all cite a same document; A cocited relation refer to that several documents are cited together by an another document. In many related work (e.g. [18]) on directed graphs, these two kinds of document co-occurrences are used to infer the similarity among documents. Probably the most well recognized way to represent the similarities among the nodes of a graph is associated with the graph Laplacian [2], say $\mathcal{L} \in \mathbb{R}^{|\mathcal{D}| \times |\mathcal{D}|}$, which is defined as:

$$\mathcal{L} = I - \alpha S_d, \tag{6}$$

where S_d is the similarity matrix on directed graphs as measured in Eq. 5; $\alpha \in (0, 1)$ is a parameter for the Laplacian to be invertible; I is an identity matrix. Note that S_d is symmetric and positive-semidefinite.

Next we give the method to learn from G_D.

Objective function: Suppose we have a document embedding $X = [\mathbf{x}_1, ...\mathbf{x}_k]$ where $\mathbf{x}_i$ contains the distribution of values of all documents on the i-th dimension of a k-dimensional latent space. The overall "lack-of-smoothness" of the distribution of these vectors w.r.t. to the Laplacian $\mathcal{L}$ can be measured as

$$\Omega(X) = \sum_{1 \le i \le k} \mathbf{x}_i^T \mathcal{L} \mathbf{x}_i = \mathrm{Tr}(X^T \mathcal{L} X), \tag{7}$$

where $X = [\mathbf{x}_1, ...\mathbf{x}_k]$. Here we seek to minimize the overall "lack-of-smoothness" so that the relative positions of documents in X will reflect the similarity in S_d.

Constraint: In addition to the objective function of X, we enforce a constraint on X so as to avoid getting a trivial solution (Note that $X = 0$ minimizes Eq. 7 if there is no constraint on X). We choose to use the newly proposed log-determinant heuristic on $X^T X$, a.k.a the log-det heuristic, denoted by $\log |X^T X|$ [6]. It has been shown that the $\log |Y|$ is a smooth approximation for the rank of Y if Y is a positive semidefinite matrix. It is obvious the gram matrix $X^T X$ is

[3] In practice when some nodes have no outgoing or incoming edges, the probability distribution over nodes can incorporate certain randomness so that P denotes an ergodic Markov chain.

positive semidefinite. Thus, when we maximize $\log |X^T X|$, we effectively maximize the rank of X, which is at most k. Another way to understand $\log |X^T X|$ is to note that $|X^T X| = \prod_i \lambda_i(X^T X) = \prod_i \sigma_i(X)^2$, where $\lambda_i(Y)$ is the i-th eigen-value of Y and $\sigma_i(X)$ is the i-th singular value of X. Therefore, a full-ranked X is preferred when $\log |X^T X|$ is maximized. For more reasons on using the log-det heuristic, refer to the *Comments* below and [6].

Using the log-det heuristic, we arrive at the combined optimization problem:

$$\min_X \left\{ \mathrm{Tr}(X^T \mathcal{L} X) - \log |X^T X| \right\} \tag{8}$$

where $\mathrm{Tr}(A)$ is the trace function defined as the sum of diagonal elements of A. It has been shown that $\max\{\log |X^T X|\}$ (or equivalently $\min\{-\log |X^T X|\}$) is a convex problem [6]. So Eq. 8 is still a convex problem.

Comments: First, it is interesting to notice that we did not use the traditional constraint on X (such as the orthonormal constraint of the subspace used in PCA [15]). The reason of choosing log-det heuristic in our case is because that (1) the orthonormal constraint is non-convex while the remaining of the problem is; (2) the orthonormal constraint cannot be solved by gradient-based methods and thus cannot be efficiently solved and cannot be easily combined with the other two factorizations in the following sections; (3) the log-det, $\log |X^T X|$, has a small problem scale ($k \times k$) and can be solved effectively by gradient-based methods. Second, note a key difference of this work from related work on link matrix factorization (e.g. [20]) is that we seek to determine X to comply with the graph Laplacian (not to factorize the link matrix) which gives us a convex problem that is global optimal.

3.2 Learning from Author Matrix: A

Here, we show how to learn from an author matrix, A, which is the adjacency matrix of the bipartite graph, G_{DA}, that captures the relationship between $\mathcal{D}$ and $\mathcal{A}$. We can use G_{DA} to encode two kinds of information between authors and documents, one being the authorship and the other being the author-citation-ship. To encode authorship, we let $A \in \mathbb{I}^{|\mathcal{D}| \times |\mathcal{A}|}$ ($\mathbb{I} \in \{0, 1\}$), where $A_{i,j}$ indicates whether the i-th paper is authored by the j-th author; To encode author-citation-ship, we assume $A \in \mathbb{R}^{|\mathcal{D}| \times |\mathcal{A}|}$, where $A_{i,j}$ can be the number of times that document i is cited by author j (or the logarithm of the citation count for rescaling).

We consider both kinds of author-document relationship equivalently using matrix factorization, where authors in both cases are considered social features of documents, inferring similarities between documents. The basic intuition is that the document related to a same set of authors should be relatively close in the latent space X. The inference of this intuition to citation recommendation is that the other work of an author will be recommended given a reader is interested in several work by similar authors.

Given the authorship matrix $A \in \mathbb{R}^{|\mathcal{D}| \times |\mathcal{A}|}$, we want to use X to approximate it. Let the authors be described by an author profile matrix $W \in \mathbb{R}^{|\mathcal{A}| \times k}$. We can approximate A by XW^T as:

$$\min_{X,W} \|A - XW^T\|_F^2 + \lambda_1 \|W\|_F^2, \tag{9}$$

where X and W are the minimizers. To prevent overfitting, the second term is used, where λ_1 is the parameter. Note that later we will combine Eq. 8 and Eq. 9; So we do not show the constraint on $\|X\|_F^2$ here. It is worth mentioning that the idea of using two latent semantic spaces to approximate a co-occurrence matrix is similar to that used in document content analysis (e.g. the LSA [5]).

3.3 Learning from Venue Matrix: V

In the above, we have given the method for learning a representation of $\mathcal{D}$ from a directed citation graph G_D and an undirected bipartite graph G_{DA}. In this section, we are given an additional piece of categorical information, which can be described by the bipartite venue graph G_{DV}, where one set of nodes are the documents from $\mathcal{D}$ and the other set are the venues from $\mathcal{V}$.

Similar to A, we have the venue matrix $V \in \mathbb{I}^{|\mathcal{D}| \times |\mathcal{V}|}$, where $V_{i,j}$ denotes whether document i is in venue j. However, a key difference here is that each row in V has at most one nonzero element because one document can proceed in at most one venue. Although we could as well employ XW^T to approximate V (as in Sec. 3.2), we will show that the special property of V can help us cancel the variable matrix W, and thus reducing the optimization problem size for better efficiency. Accordingly, we follow a similar but different approach. In particular, let us consider to use V to predict the X via linear combinations. Suppose we have W_2 as the coefficient, we seek to minimize the following:

$$\min_{X,W_2} \|VW_2^T - X\|_F^2. \tag{10}$$

One can understand Eq. 10 in this way: Here each column of W_2 can be considered as a cluster center of the corresponding class (i.e., the venues). Then solving Eq. 10 in fact simultaneously (1) pushes the representation of documents close to their respective class centers; and (2) optimizes the centers to be close to their members.

Next, we cancel W_2 using the unique property of our venue matrix V. Setting the derivative to be zero, we have $0 = \partial \|VW_2^T - X\|_F^2 / \partial W_2 = 2(V^T V W_2 - V^T X)$, suggesting that $W_2 = (V^T V)^{-1} V^T X$. Note that $V^T V$ is diagonal matrix and is thus invertible. Plug in W_2 back to Eq. 10. We arrive at the optimization where W_2 is canceled:

$$\min_X \|V(V^T V)^{-1} V^T X - X\|_F^2, \tag{11}$$

where $(V^T V)^{-1} V^T$ is the pseudo inverse of V. Here since $V^T V$ is $|\mathcal{V}| \times |\mathcal{V}|$ diagonal matrix, its inverse can be computed in $|\mathcal{V}|$ flops. Meanwhile, $V(V^T V)^{-1} V^T$ is block diagonal where each block denotes a complete graph among all documents within the same venue. Note that Eq. 9 cannot be handled in the same way because $(A^T A)^{-1}$ is a dense matrix, resulting in a $|\mathcal{D}| \times |\mathcal{D}|$ dense matrix of $A(A^T A)^{-1} A^T$, which in practice raises scalability issues.

3.4 Learning Document Embedding

We have arrived at a combined optimization formulation given the above sub-problems. We will combine Eq. 8, Eq. 9 and Eq. 10 in a unified optimization framework. Define the new objective $J(X, W)$ as a function of X, W. We have an optimization below to learn the document embedding matrix X:

$$\begin{aligned} J(X, W) = \quad & (\mathrm{Tr}(X^T \mathcal{L} X) - \log |X^T X| \\ & + \alpha \|A - XW^T\|_F^2 + \lambda \|W\|_F^2 \\ & + \beta \|V(V^T V)^{-1} V^T X - X\|_F^2) \end{aligned} \tag{12}$$

where λ is the weight of regularization on W; α is the weight for learning from A; β is the weight for learning from V. In this paper, we only empirically find the best values for α and β that yield the best F-scores for the current data set. Future work on how to choose parameter values will be helpful to practitioners.

The optimization illustrated above can be solved using standard Conjugate Gradient (CG) method, where the key step is the evaluation of objective function and the gradient. In Appendix .1, we show the gradients for the combined optimization.

After X is calculated, we can use linear model in the recommendation, i.e. $f^* = X(X^T X)^{-1} X^T y$. We can obtain efficiency advantage over the power method as in Eq. 3.

4. INCREMENTAL UPDATE OF DOCUMENT EMBEDDING

An incremental version of our new method will be proposed in this section. The goal of incremental update of X is to avoid heavy computation of known documents when there is a small size of update. The incentive for designing an incremental update algorithm is to delay (or avoid) recomputation in a batch approach. The incremental update of X we will give is an efficient approximate solution. In particular, suppose we have used document $\mathcal{D}_0$, $\mathcal{V}_0$, $\mathcal{A}_0$ and their relationship at time t_0 to compute a document embedding X_0 for the document set $\mathcal{D}_0$. Now, at time t_1, we have observed an additional set of new documents $\mathcal{D}_1$. How can we use the pre-computed X_0 to compute an embedding of $\mathcal{D}_1$ in $X_1 \in \mathbb{R}^{|\mathcal{D}_1| \times k}$ efficiently? Note that typically $|\mathcal{D}_1|$ is much smaller than $|\mathcal{D}_0|$.

4.1 Rewriting Objective Functions

We rewrite the objective function in Eq. 12. Let X be the minimizer. We assume that the embedding of old documents is in X_0 and the $X_1 \in \mathbb{R}^{|\mathcal{D}_1| \times k}$ is the embedding for $\mathcal{D}_1$. Here $X^T = [X_0^T, X_1^T]$. Let the updated three graphs be encoded in the three new matrices below:

$$A = \begin{bmatrix} A_{00} & A_{01} \\ A_{10} & A_{11} \end{bmatrix}, V = \begin{bmatrix} V_0 \\ V_1 \end{bmatrix}, \mathcal{L} = \begin{bmatrix} L_{00} & L_{01} \\ L_{01}^T & L_{11} \end{bmatrix},$$

where the A encodes the new document-author relationship; the V encodes the new the document-venue relationship (assuming no emergence of new venues); and the $\mathcal{L}$ denotes the new Laplacian calculated on the updated document citation graph. By convention, the index 0 corresponds to the original part of the matrix and the index 1 indicates the new part. For example, V_0 is the venue matrix at time t_0 and V_1 is the venue matrix at time t_1.

Consider the objective function in Eq. 12. After several rewrites as entailed in Appendix .2, the objective function in Eq. 12 on the new set of matrices now becomes the following:

$$\begin{aligned} J = \quad & \mathrm{Tr}(X_0^T L_{00} X_0 + 2 X_0^T L_{01} X_1 + X_1^T L_{11} X_1) \\ & - \log |X_0^T X_0 + X_1^T X_1| \\ & + \lambda \|W_0\|_F^2 + \lambda \|W_1\|_F^2 \\ & + \alpha \|A_{00} - X_0 W_0^T\|_F^2 + \alpha \|A_{01} - X_0 W_1^T\|_F^2 \\ & + \alpha \|A_{10} - X_1 W_0^T\|_F^2 + \alpha \|A_{11} - X_1 W_1^T\|_F^2 \\ & + \beta \|(V_0 \Sigma^{-1} V_0^T - I) X_0 + V_0 \Sigma^{-1} V_1^T X_1\|_F^2 \\ & + \beta \|V_1 \Sigma^{-1} V_0^T X_0 + (V_1 \Sigma^{-1} V_1^T - I) X_1\|_F^2, \end{aligned} \tag{13}$$

where the coefficients are $\mathcal{L}$, A, V, and $\Sigma = (V_0^T V_0 + V_1^T V_1)$; The variables are $X = \begin{bmatrix} X_0 \\ X_1 \end{bmatrix}$, $W = \begin{bmatrix} W_0 \\ W_1 \end{bmatrix}$; The parameters are α, β, λ.

4.2 Efficient Approximate Solution

We will make the Eq. 13 more efficient in this section, hoping to only calculate the incremental part of X for the new documents in $\mathcal{D}_1$.

First, let us assume that the incremental update of X only seek to update the embedding of $\mathcal{D}_1$ but does not change the original embedding of $\mathcal{D}_0$, i.e. that X_0 is fixed. Similarly, W_0 is fixed for the authors observed before. Second, we can see that V_0 in V is fixed because documents will not change venues over time. Third, we show that the segment in the new Laplacian L_{01} is approximately zero because no old documents can cite new documents which results in relatively small stationary probabilities on the new documents (we will show more details for this proposition in Appendix .3). Given the above assumptions and observations, after discarding the constant terms, we have the following optimization for incremental update of X:

$$\begin{aligned} J_{app} = \quad & \mathrm{Tr}(X_1^T L_{11} X_1) - \log |X_0^T X_0 + X_1^T X_1| \\ & + \alpha \|A_{01} - X_0 W_1^T\|_F^2 + \alpha \|A_{10} - X_1 W_0^T\|_F^2 \\ & + \alpha \|A_{11} - X_1 W_1^T\|_F^2 + \lambda \|W_1\|_F^2 \\ & + \beta \|(V_0 \Sigma^{-1} V_0^T - I) X_0 + V_0 \Sigma^{-1} V_1^T X_1\|_F^2 \\ & + \beta \|V_1 \Sigma^{-1} V_0^T X_0 + (V_1 \Sigma^{-1} V_1^T - I) X_1\|_F^2, \end{aligned} \tag{14}$$

where $\Sigma = (V_0^T V_0 + V_1^T V_1)$. The variables are X_1 and W_1 that has $|\mathcal{D}_1| \times k$ and $|\mathcal{A}_1| \times k$ elements respectively. Since $\mathcal{D}_1$ and $\mathcal{A}_1$ are very small, the incremental calculation of X_1 can be achieved very efficiently. Again, this problem can be solved using conjugate gradient method where the gradients of Eq. 14 are presented in Appendix .1.

5. EXPERIMENTS

A real-world data set for experimentation was generated by sampling documents from CiteSeer using combined document meta-data from CiteSeer and another two sources (the ACM Guide, http://portal.acm.org/guide.cfm, and the DBLP, http://www.informatik.uni-trier.de/ ley/db) for enhanced data accuracy and coverage. The meta-data was processed so that the ambiguous author names and noisy venue titles were canonicalized [4]. Since the data in CiteSeer are collected automatically by crawling the Web, we may not have enough information about certain authors. Accordingly, we collected the documents by those top authors in CiteSeer ranked by their numbers of documents. Then we collected the venues of these documents. Similarly, we kept those venues with most documents in the prepared subset and discarded the venues that include fewer documents. Following the same procedure, two datasets were prepared with different sizes. The first dataset, referred to as DS_1, has 400 authors, $9,197$ documents, 50 venues, and $19,844$ citations; The second dataset, referred to as DS_2, which is larger in size, has 800 authors, $15,073$ documents, 100 venues, and $38,614$ citations.

[4] Venues with only temporal differences, such as the conference proceedings from different years or the journals of different issues, were treated as the same venue.

5.1 Evaluation Metrics

The performance of recommendation can be measured by a wide range of metrics, including user experience studies and click-through monitoring. For experimental purpose, this paper will evaluate the proposed method against citation records by cross-validation. In particular, we randomly remove t documents, use the remaining documents as the seeds, perform recommendations, and judge the recommendation quality by examining how well these removed documents can be retrieved. As suggested by real user usage patterns, we are only interested in the top recommended documents. Quantitatively, we define the recommendation *precision* (p) as the percentage of the top recommended documents that are in fact from the true citation set. The *recall* (r) is defined as the percentage of true citations that are really recommended in the top m documents. The *F-score*, which combines *precision* and *recall* is defined as $f = (1 + \delta^2)rp/(r + \delta^2 p)$, where $\delta \in [0, \infty)$ determines how relatively important we want the recall to be (Here we use $\delta = 1$, i.e. F-1 score, as in many related work.) [5]. We have introduced a parameter in evaluation, m, which is the number of top documents we evaluate the f-score at.

5.2 Recommendation Quality

This section introduces the experiments on recommendation quality. We compare the recommendation by our algorithm with two other baselines: one based on Laplacian on directed graphs [2] and label propagation using graph Laplacian [18] (named as *Lap*) and the other based on Singular Vector Decomposition of the author matrix (named as *SVD*) [6]. We chose to compare with the *Lap* method to see whether the fusion of different graphs can effectively produce additional information than the original graph citation graph; We chose the SVD on author matrix as another baseline because we would like compare our method against the traditional CF method on the additional graph information (as one can argue that the significant improvement of the new method is purely due to the use of the additional information).

Table 1 and Table 2 list the f-scores (defined in Sec. 5.1) of three different methods (our new method with *Lap* and *SVD*) on two datasets (DS_1 and DS_2). Table 1 for different number of top documents evaluated on (denoted by m). We are able to see that the new method outperforms both *Lap* and SVD significantly on both datasets in different settings of parameters. In general, the new method are $3 - 5$ times better in f-score than *Lap* and 2.5 times better than SVD. The *Lap* method under-performs SVD on the very top documents but beats it if evaluated on more top documents. In addition, we notice that the f-scores get better in general as we look at more top documents. Also, the f-scores on the smaller dataset DS_1 are generally higher than those on the larger dataset DS_2. Here, we can see that the recommendation quality can be significantly improved by using the author matrix as the additional information. Note that the different information, when used individually, such as the *Lap* on the citation graph or the SVD on the author graph, can be not as good. However, if the multiple information are combined, the performance is greatly improved[7].

	f \ m	**m=t**	**m=5**	**m=10**
$DS1$	f(lap)	0.013	0.048	0.192
	f(svd)	0.035	0.086	0.138
	f(new)	**0.108**	**0.242**	**0.325**
$DS2$	f(lap)	0.011	0.046	0.156
	f(svd)	0.027	0.072	0.109
	f(new)	**0.083**	**0.158**	**0.229**

Table 1: The f-score calculated on different numbers of top documents, m.

	f \ t	**t=1**	**t=2**	**t=3**	**t=4**
$DS1$	f(lap)	0.041	0.048	0.075	0.086
	f(svd)	0.062	0.088	0.099	0.103
	f(new)	**0.197**	**0.242**	**0.248**	**0.252**
$DS2$	f(lap)	0.037	0.047	0.068	0.077
	f(svd)	0.049	0.072	0.082	0.086
	f(new)	**0.121**	**0.158**	**0.181**	**0.182**

Table 2: The f-score w.r.t. different numbers of left-out documents, t.

5.3 Parameter Effect

The effect of parameters for the new method is experimented in this section. We experiment with different settings of dimensionality, or k, and weights on authors and venues, or α and β. In Table 3, we show the f-scores for different k's. It occurs that the f-scores become higher for greater k. We believe this is because the higher dimensional space can better captures the similarities in the original citation graphs. However, on the other hand, we observe that it takes longer training time for greater k. Seeking k thus become a trade-off between quality and efficiency. In our experiments, we chose $k = 100$ as greater k do not seem to give much better results. The CPU time for training at different k's are illustrated in Table 4.

Fig. 3 illustrates the f-scores for different settings of α and β, which are respectively the weights on authors and venues. We determine which of the two components obtains greater improvement if incorporated, search for the best parameter for this component, fix it, and then search for the best parameter for the other component. In our experiments, we

[5] Note that even it is the recommendation problem that we address, we cannot use the Mean Average Error (MAE), which is used for measuring the quality of a *Collaborative Filtering* algorithm, because we do not seek to approximate the ratings of documents but to preserve their preference orders in the recommendation ranking. Similarly, we cannot use Discounted Cumulated Gain (DCG), which is used for evaluating a rank list, because the numbers of true citations in each prediction task are different.

[6] If we consider the author matrix as a user-item rating matrix, the SVD of the rating is in fact a simple *Collaborative Filtering* (CF) method. However, due to different objectives of our problem and the traditional CF, we will see later that our method outperforms SVD towards our goal significantly.

[7] In our experiments, additionally, we work with different methods of formulating the author matrix, A, for example, using the number of citations from authors to documents in A. The experiments show that using the citation-ship in A can be even better. Due to space limit, here we present the experiments with authorship in A only.

f \ k	k=50	k=100	k=150	k=200
$DS1$	0.203	0.242	0.249	0.262
$DS2$	0.095	0.158	0.181	0.197

Table 3: The f-score w.r.t. different setting of dimensionality, k.

	t(lap)	t(new)			
time \ k		k=50	k=100	k=150	k=200
$DS1$	694s	**440s**	**502s**	**558s**	**621s**
$DS2$	940s	**638s**	**743s**	**820s**	**910s**

Table 4: The CPU time for recommendations w.r.t. different dimensionality, k.

observe that adding the author component tends to improve the recommendation quality better so we first tune α, which yields different f-scores, as shown by the blue curve in Fig. 3. Then we fix the $\alpha = 0.1$ and tune β, arriving at the best f-score at $\beta = 0.05$.

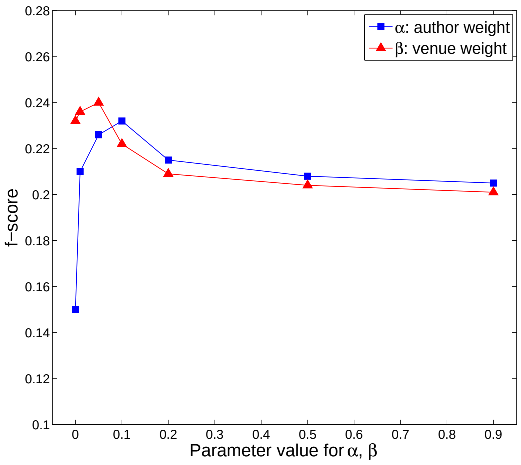


Figure 3: f-scores for different settings of weights on the authors, α, and on the venues, β. The α is tuned first for $\beta = 0$; Then β is tuned for the fixed best $\alpha = 0.1$.

5.4 Incremental Update

Here we present the experiments for another contribution of this work: *incremental update*. The *incremental update* method we propose seeks to determine an approximate embedding of documents by working with the incremental data and the relationship between the new data and the old. We evaluate this new method in its training time, recommendation quality, and propagation of errors.

Fig. 4 illustrates the comparison of training time for the incremental method and batch update by percentage on both datasets. We try to use a fair baseline. In particular, we compare with a percentage of batch update time, where the percentage reflects the relative amount of incremental data. As illustrated in Fig. 4, the incremental method takes on average $1/2 - 1/5$ of the training time of batch method. The improvement is more significant on larger dataset (DS_2) than small ones (DS_1).

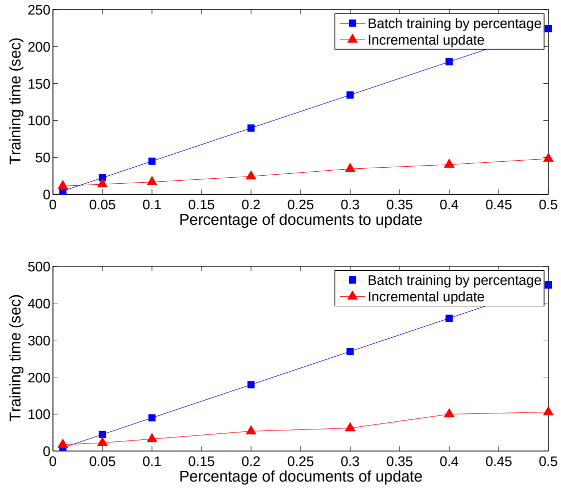


Figure 4: Training time for incremental update and batch method w.r.t. different percentage of incremental data on $DS1$ and $DS2$. The training time for batch method is the corresponding percentage of the overall training time.

The next natural question to ask is how much quality has to be compromised for the improvement of efficiency. Fig. 5 present the comparison of f-scores for different percentage of incremental using the incremental method with the batch method applied to the full data. It turns out that the performance of incremental method deteriorates as the incremental data takes a large percentage. Fortunately, the f-scores decrease at a slower ratio for the larger dataset (DS_2). This is because that more information is captured by the larger dataset with larger absolute size. On average, the deterioration of recommendation quality can be significant if the incremental data takes more than 30% of the data. So we would suggest re-run the batch process when the updated corpus exceeds the original size significantly.

Finally, we present the propagation of errors if the incremental update is applied to multiple times. It has come to our attention that the performance deteriorates at a faster pace if one applies multiple steps of incremental updates. Fig. 5.4 illustrates the f-scores w.r.t. different numbers of steps in the incremental updates, for different overall percentage to update. Notice that the f-scores deteriorate faster if the overall percentage of update is greater. Also, the f-scores decrease slower at first $1 - 2$ steps and faster from the $3rd$ step onwards. It is then suggested that the new incremental method should be used with caution, preferring fewer number of uses or on a larger percentage of data for each use. It seems that the error in the incremental updates is propagated more than linearly.

The incremental update methods presented in this section addresses the scalability issues in recommendation of large-scale dataset on the Web. In practice, we recommend a combination of batch update and incremental update seeking a tradeoff between efficiency and scalability.

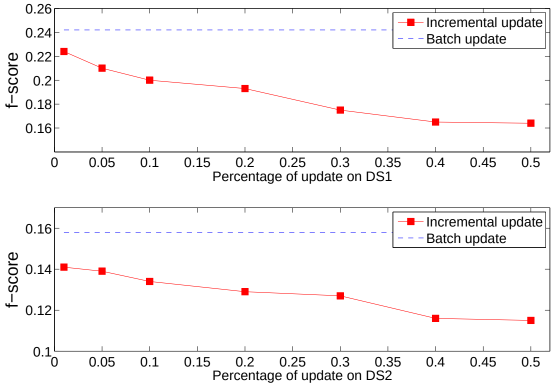


Figure 5: f-scores for different percentage of incremental data in the incremental update, on $DS1$ and $DS2$, w.r.t. the batch method applied to the full data.

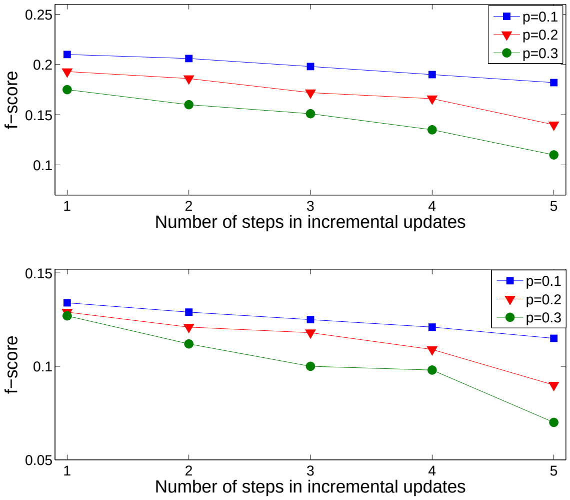


Figure 6: f-scores for different numbers of steps in the incremental updates, for different overall percentage (p) to update, on $DS1$ and $DS2$.

6. RELATED WORK

This work is first related to a family of work on categorization of networked documents. Categorization of networked documents is developed based on the link structure and the co-citation patterns (e.g. [8] for Web document clustering). In [8], all links are treated as undirected edge of the link graph and the content information is only used for weighing the links by the textual similarity between documents on both ends of the link. Very recently, Chung[2] has proposed a regularization framework on directed graphs. Soon after, Zhou et.al. [18] used this regularization framework on directed graphs for semi-supervised learning, which also seek to explain ranking and categorization in the same semi-supervised learning framework. Later, a work by Zhou et.al. extended the regularization to multiple graphs with the same set of vertices [16], which, however, is different from this work where the item set can be either a full set or a subset of the graphs in question.

This work also relates to the category of work that approach document analysis via embedding documents onto a relatively low dimensional latent space [5, 17]. Latent Semantic Indexing (LSI) [5] is a representative work in this category that uses a latent semantic space to implicitly capture the information of documents. Analysis tasks, such as classification, could be performed on the latent space. Another commonly used method, Singular Value Decomposition (SVD), ensures that the data points in the latent space can optimally reconstruct the original documents. Based on similar idea, Hofmann [9] proposed a probabilistic model, called Probabilistic Latent Semantic Indexing (pLSI). This work is similar but different to pLSI in that we not only approximate one single document matrix but several matrices at the same time.

Finally, this work shares the idea of related work on combining multiple sources of information. In this category, prior work by Cohn and Hofmann [4] extends the latent space model to construct the latent space from both content and link information, using content analysis based on pLSI and PHITS [3], which is a direct extension of pLSI on the links. In PLSI+PHITS, the link is constructed with the linkage from the topic of the source web page to the destination web page. In that model, the outgoing links of the destination web page have no effect on the source web page. In other words, the overall link structure is not utilized in PHITS. Communitiy discovery has also been done purely based on document content [19]. Recent work. [20] utilizes the overall link structure by representing links using the latent information of their both end nodes. In this way, the latent space truly unifies the content and the underlying link structure. Our work is similar to that of [20] but we not only considers links but also co-link patterns by using the Laplacian on directed graphs.

7. CONCLUSIONS AND FUTURE WORK

We address the item-based collaborative filtering problem for items that are networked. We propose a new method for combining multiple graphs in order to measure item similarities. In particular, the new method seeks a single low-dimensional embedding of items that captures the relative similarities among them in the latent space. We formulate this as an optimization problem, where the learning of three general types of graphs are formulated as three subproblems, each using a factorization strategy tailored to the unique characteristics of the graph type. Based on the obtained item embedding, a new recommendation framework is developed using semi-supervised learning on graphs. In addition, we address the scalability and propose an incremental version of the new method. Approximate embeddings are calculated only for new items making it very efficient. The new batch and incremental methods are evaluated on two real world datasets prepared from CiteSeer. Experiments have demonstrated significant quality improvement for our batch method and significant efficiency improvement with tolerable quality loss for our incremental method. For future work, we will pursue other applications of the new graph fusion technique, such as clustering or classification. In addition, we want to extend our framework to graphs with hyperedges.

8. ACKNOWLEDGMENTS

This work was funded in part by the National Science Foundations and the Raytheon Corporation.

APPENDIX

.1 The Gradients for Eq. 12 and Eq. 14

The gradients for Eq. 12 are:

$$\begin{aligned}\frac{\partial J}{\partial X} &= 2\mathcal{L}X - 2X(X^TX)^{-1} \\ &\quad +2\alpha(XW^TW - AW) + \\ &\quad +2\beta(VV^\dagger - I)^T(VV^\dagger - I)X \qquad (15)\\ \frac{\partial J}{\partial W} &= 2\alpha(WX^TX - A^TX) + 2\lambda W \qquad (16)\end{aligned}$$

where $V^\dagger = (V^TV)^{-1}V^T$ is the pseudo inverse of V. When searching for the solutions, we vectorize the gradients of X, W into a long vector. In implementation, different calculation order of matrix product leads to very different efficiency. For example, it is much more efficient to calculate $(VV^\dagger - I)^T(VV^\dagger - I)X$ as $(V^\dagger)^TV^TVV^\dagger X - 2VV^\dagger X + X$ because V and $V^\dagger$ are very sparse.

The gradients for Eq. 14 are:

$$\begin{aligned}\frac{1}{2}\frac{\partial J_{app}}{\partial X_1} &= L_{11}X_1 - X_1(X_0^TX_0 + X_1^TX_1)^{-1} \\ +\alpha(X_1W_0^TW_0 &- A_{10}W_0) + \alpha(X_1W_1^TW_1 - A_{11}W_1) \\ &+\beta(Q_0^TP_0 + Q_0^TQ_0X_1) + \beta(Q_1^TP_1 + Q_1^TQ_1X_1) \qquad (17)\end{aligned}$$

where $P_0 = (V_0\Sigma^{-1}V_0^T - I)X_0$, $Q_0 = V_0\Sigma^{-1}V_1^T$, $P_1 = V_1\Sigma^{-1}V_0^TX_0$, $Q_1 = (V_1\Sigma^{-1}V_1^T - I)$, $\Sigma = (V_0^TV_0 + V_1^TV_1)$. The gradients of Eq. 14 w.r.t. W_1 are

$$\begin{aligned}\frac{1}{2}\frac{\partial J_{app}}{\partial W_1} &= \alpha(W_1X_0^TX_0 - A_{01}^TX_0) \\ &\quad +\alpha(W_1X_1^TX_1 - A_{11}^TX_1) + \lambda W_1 \qquad (18)\end{aligned}$$

.2 Rewriting the Objective Functions

First, for the terms in Eq. 12 for learning from $\mathcal{A}$, we introduce another set of variables in $W_1 \in \mathbb{R}^{|\mathcal{A}_1|\times k}$, to describe the new authors in $\mathcal{A}_1$. Then we have $W^T = [W_0^T, W_1^T]$. Let the document-author relationship be encoded in the author matrix A: $A = \begin{bmatrix} A_{00} & A_{01} \\ A_{10} & A_{11}\end{bmatrix}$. Then we have the following:

$$\begin{aligned}\|A - XW^T\|_F^2 &= \left\| \begin{bmatrix} A_{00} & A_{01} \\ A_{10} & A_{11}\end{bmatrix} - \begin{bmatrix} X_0 \\ X_1 \end{bmatrix}\begin{bmatrix} W_0^T & W_1^T\end{bmatrix}\right\|_F^2 \\ &= \left\| \begin{matrix} A_{00} - X_0W_0^T & A_{01} - X_0W_1^T \\ A_{10} - X_1W_0^T & A_{11} - X_1W_1^T \end{matrix}\right\|_F^2 \\ &= \|A_{00} - X_0W_0^T\|_F^2 + \|A_{01} - X_0W_1^T\|_F^2 \\ &\quad +\|A_{10} - X_1W_0^T\|_F^2 + \|A_{11} - X_1W_1^T\|_F^2. \quad (19)\end{aligned}$$

and $\|W\|_F^2$ becomes

$$\|W\|_F^2 = \left\| \begin{matrix} W_0 \\ W_1 \end{matrix} \right\|_F^2 = \|W_0\|_F^2 + \|W_1\|_F^2. \qquad (20)$$

Second, for the term in Eq. 12 regarding learning from venue matrix, V, we assume that there are no new venues showing up between t_0 and t_1. So the new V takes the form as $V = \begin{bmatrix} V_0 \\ V_1 \end{bmatrix}$ where V_0 is the venue matrix at time t_0. Let the component $V(V^TV)^{-1}V^T = \Phi$. We can see that the learning objective becomes

$$\|(\Phi - I)X\|_F^2, \qquad (21)$$

where

$$\begin{aligned}\Phi &= \begin{bmatrix} V_0 \\ V_1 \end{bmatrix}\left(\begin{bmatrix} V_0^T & V_1^T\end{bmatrix}\begin{bmatrix} V_0 \\ V_1 \end{bmatrix}\right)^{-1}\begin{bmatrix} V_0^T & V_1^T\end{bmatrix} \\ &= \begin{bmatrix} V_0 \\ V_1 \end{bmatrix}(V_0^TV_0 + V_1^TV_1)^{-1}\begin{bmatrix} V_0^T & V_1^T\end{bmatrix} \\ &= \begin{bmatrix} V_0\Sigma^{-1}V_0^T & V_0\Sigma^{-1}V_1^T \\ V_1\Sigma^{-1}V_0^T & V_1\Sigma^{-1}V_1^T \end{bmatrix} \qquad (22)\end{aligned}$$

and $\Sigma = (V_0^TV_0 + V_1^TV_1)$ is a diagonal matrix whose inverse is very easy to compute. Then we plug Eq. 22 into Eq. 21. After several simple manipulations, we arrive at the following learning objective for venues:

$$\begin{aligned}&\|(V_0\Sigma^{-1}V_0^T - I)X_0 + V_0\Sigma^{-1}V_1^TX_1\|_F^2 + \\ &\quad \|V_1\Sigma^{-1}V_0^TX_0 + (V_1\Sigma^{-1}V_1^T - I)X_1\|_F^2, \qquad (23)\end{aligned}$$

where $\Sigma = (V_0^TV_0 + V_1^TV_1)$.

Third, for the Laplacian terms in Eq. 12 for learning from the citation graph D, we have the following identities:

$$\begin{aligned}&\mathrm{Tr}(X^T\mathcal{L}X) \\ &= [X_0^T, X_1^T]\begin{bmatrix} L_{00} & L_{01} \\ L_{01}^T & L_{11}\end{bmatrix}\begin{bmatrix} X_0 \\ X_1 \end{bmatrix} \\ &= \mathrm{Tr}(X_0^TL_{00}X_0 + 2X_0^TL_{01}X_1 + X_1^TL_{11}X_1) \qquad (24)\end{aligned}$$

and

$$\log|X^TX| = \log|X_0^TX_0 + X_1^TX_1| \qquad (25)$$

where L_{00} is a $|\mathcal{D}_0| \times |\mathcal{D}_0|$ matrix for the graph on $\mathcal{D}_0$; L_{01} is a $|\mathcal{D}_0| \times |\mathcal{D}_1|$ matrix for interaction between $\mathcal{D}_0$ and $\mathcal{D}_1$; and L_{11} is a $|\mathcal{D}_1| \times |\mathcal{D}_1|$ matrix for the graph on $\mathcal{D}_1$.

.3 The Laplacian $\mathcal{L}$ is almost block diagonal:

Here we will show that the Laplacian $\mathcal{L}$ on the new matrix D at time t_1 is near block diagonal. Recall that the citation matrix D at time t_1 can be written as $D = \begin{bmatrix} D_{00} & D_{01} \\ D_{10} & D_{11}\end{bmatrix}$. Here, D_{00} is the same as the citation matrix used to compute the Laplacian at time t_0. Remember that $\mathcal{L} = I - \alpha S$, where S is the similarity measured on the directed graph D in Eq. 5:

$$S = \frac{1}{2}(\bar{S} + \bar{S}^T) \qquad (26)$$

where $\bar{S} = \Phi^{1/2}P\Phi^{-1/2}$ and P is the stochastic matrix normalized from D and Φ is a diagonal matrix containing the stationary probabilities on each random walk state. We rewrite $\bar{S}$ as follows:

$$\bar{S} = \left[\begin{array}{cc} \Phi_{00} & \\ & \Phi_{11} \end{array} \right]^{\frac{1}{2}} \left[\begin{array}{cc} P_{00} & P_{01} \\ P_{10} & P_{11} \end{array} \right] \left[\begin{array}{cc} \Phi_{00} & \\ & \Phi_{11} \end{array} \right]^{-\frac{1}{2}} \tag{27}$$

where $P_{00}, P_{01}, P_{10}, P_{11}$ are normalized from $D_{00}, D_{01}, D_{10}, D_{11}$ and the diagonal matrix Φ_{00} (Φ_{11}) contains the stationary probabilities on the old (new) documents.

We further know that $P_{01} = 0$ because new documents $\mathcal{D}_1$ cannot be cited by the old documents. So we have:

$$\bar{S} = \left[\begin{array}{cc} \Phi_{00}^{\frac{1}{2}} P_{00} \Phi_{00}^{-\frac{1}{2}} & 0 \\ \Phi_{11}^{\frac{1}{2}} P_{10} \Phi_{00}^{-\frac{1}{2}} & \Phi_{11}^{\frac{1}{2}} P_{11} \Phi_{11}^{-\frac{1}{2}} \end{array} \right]. \tag{28}$$

And we also know that the new documents $\mathcal{D}_1$, with few citations among themselves, mainly cite the old documents in $\mathcal{D}_0$. Thus, in the case when $\mathcal{D}_0$ is much larger than $\mathcal{D}_1$, the stationary probabilities on the new documents $\mathcal{D}_1$ are very small, i.e. $\Phi_{11} \sim 0$. This gives us $\Phi_{11}^{\frac{1}{2}} P_{10} \Phi_{00}^{-\frac{1}{2}} \sim 0$. So we have shown that $\bar{S}$ is almost diagonal. Let us rewrite $\bar{S}$ as: $\bar{S} \sim diag(\Phi_{00}^{\frac{1}{2}} P_{00} \Phi_{00}^{-\frac{1}{2}}, \Phi_{11}^{\frac{1}{2}} P_{11} \Phi_{11}^{-\frac{1}{2}})$. Since $\mathcal{L} = I - \alpha S$ where $S = \frac{1}{2}(\bar{S} + \bar{S}^T)$, we know that the new $\mathcal{L} \sim \left[\begin{array}{cc} L_{00} & 0 \\ 0 & L_{11} \end{array} \right]$ which is almost block diagonal, i.e. $L_{01} \sim 0$. However, note that L_{11} is not necessarily zero because $\Phi_{11}^{\frac{1}{2}} P_{11} \Phi_{11}^{-\frac{1}{2}}$ contains both $\Phi_{11}^{\frac{1}{2}}$ and $\Phi_{11}^{-\frac{1}{2}}$. Also, note that we do not claim that L_{00} in the new $\mathcal{L}$ is identical to the original Laplacian on $\mathcal{D}_0$. Nevertheless, we discard the term $X_0^T L_{00} X_0$ in Eq. 13 because X_0 is assumed to be unchanged.

A. REFERENCES

[1] F. Chung. *Spectral Graph Theory.* American Mathematical Society, 1997.

[2] F. Chung. Laplacians and the cheeger inequality for directed graphs. *Annals of Combinatorics*, 9, 2005.

[3] D. Cohn and H. Chang. Learning to probabilistically identify authoritative documents. *Proc. ICML 2000. pp.167-174.*, 2000.

[4] D. Cohn and T. Hofmann. The missing link - a probabilistic model of document content and hypertext connectivity. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 430–436. MIT Press, 2001.

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

[6] M. Fazel, H. Hindi, and S. P. Boyd. Log-det heuristic for matrix rank minimization with applications to hankel and euclidean distance matrices. In *Proceedings of American Control Conference*, 2003.

[7] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 403–412, New York, NY, USA, 2004. ACM Press.

[8] X. He, H. Zha, C. H.Q. Ding, and H. D. Simon. Web document clustering using hyperlink structures. *Computational Statistics & Data Analysis*, 41(1):19–45, November 2002.

[9] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1-2):177–196, 2001.

[10] T. Hofmann. Latent semantic models for collaborative filtering. *ACM Trans. Inf. Syst.*, 22(1):89–115, 2004.

[11] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. Item-based collaborative filtering recommendation algorithms. In *WWW '01: Proceedings of the 10th international conference on World Wide Web*, pages 285–295, New York, NY, USA, 2001. ACM Press.

[12] F. Wang, S. Ma, L. Yang, and T. Li. Recommendation on item graphs. In *ICDM '06: Proceedings of the Sixth International Conference on Data Mining*, pages 1119–1123, Washington, DC, USA, 2006. IEEE Computer Society.

[13] J. Wang, A. P. de Vries, and M. J. T. Reinders. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 501–508, New York, NY, USA, 2006. ACM Press.

[14] K. Yu, A. Schwaighofer, V. Tresp, X. Xu, and H.-P. Kriegel. Probabilistic memory-based collaborative filtering. *IEEE Transactions on Knowledge and Data Engineering*, 16(1):56–69, 2004.

[15] H. Zha, C. Ding, M. Gu, X. He, and H. Simon. Spectral relaxation for k-means clustering. In *Neural Information Processing Systems*, volume 14, 2001.

[16] D. Zhou and C. J. C. Burges. Spectral clustering and transductive learning with multiple views. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, pages 1159–1166, 2007.

[17] D. Zhou, I. Councill, H. Zha, and C. L. Giles. Discovering temporal communities from social network documents. In *ICDM'07: Proceedings of the 7th IEEE International Conference on Data Mining*, 2007.

[18] D. Zhou, J. Huang, and B. Scholkopf. Learning from labeled and unlabeled data on a directed graph. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 1036–1043, 2005.

[19] D. Zhou, E. Manavoglu, J. Li, C. L. Giles, and H. Zha. Probabilistic models for discovering e-communities. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 173–182. ACM Press, 2006.

[20] S. Zhu, K. Yu, Y. Chi, and Y. Gong. Combining content and link for classification using matrix factorization. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, 2007.