Discovering Temporal Communities from Social Network Documents

Ding Zhou[1] Isaac Councill[2] Hongyuan Zha[3] C. Lee Giles[2]

Computer Science and Engineering[1]
Information Science and Technology[2]
Pennsylvania State University,
University Park, PA

College of Computing[3],
Georgia Institute of Technology,
Atlanta, GA

Abstract

This paper studies the discovery of communities from social network documents produced over time, addressing the discovery of temporal trends in community memberships. We first formulate static community discovery at a single time period as a tripartite graph partitioning problem. Then we propose to discover the temporal communities by threading the statically derived communities in different time periods using a new constrained partitioning algorithm, which partitions graphs based on topology as well as prior information regarding vertex membership. We evaluate the proposed approach on synthetic datasets and a real-world dataset prepared from the CiteSeer.

1 Introduction

Social network analysis (SNA) is an established field in sociology recently becoming popular for computer scientists, which is motivated in part by the increasing amount of personal and social information available online. Community discovery is a classical problem in social network analysis, where the goal is to discover related groups of social actors such that they are intra-group close and inter-group loose. Well known graph-theoretic methods include spectral graph partitioning [2], and clusteringbased on random walks [5]. Spectral graph partitioning is a classical spectral method based on the Laplacian of the graph adjacency matrix [2], with a characteristic focus on the design of cost functions for partitioning graphs. Random walk-based clustering described in [5] applies random walks to the graphs iteratively such that the edge weight between two vertices is modified based on the probabilities that the random walk revisits one of the vertices through the other.

Despite the wide range of choices for partitioning homogeneous networks, research on discovering communities from heterogeneous social networks is rather limited Treating heterogeneous graphs the same as homogeneous ones leads to difficulty in normalization since different edge types may be incomparable [3]. However, observations of real-world networks often indicate diverse network structures, many of which can be modeled as heterogeneous networks of social actors and the other node types such as documents (e.g. emails, blogs, collaborative publications) or social events. In this paper, we are particularly interested in communication documents as these data sources represent the most widely available sources of information regarding social networks.

Discovering communities from documents is a recent trend. Popular approaches are either content-based or graph-theoretic. One popular content-based approach is to mine information via probabilistic generative modeling, where the social actors or communities are considered as variables in the generation of document content [6, 8]. Alternatively, a graph-theoretic approach can consider the documents as an additional set of vertices connected to authors in a bipartite [7] or tripartite [3] graph structure. These methods, however, work with only a static snapshot of network data. The issues of document time and the temporal community development are generally overlooked.

This paper addresses the community discovery problem in a temporal heterogeneous social network consisting of authors, document content, and the venues in which the documents are published, all observed over time. We propose a new framework that addresses the two main challenges in this new problem: (a) handling of the heterogenous network and (b) incorporation of the temporal aspect of the data. For (a), we formulate community discovery in a heterogeneous social network (the social network is a network of authors, words, and publication venues) as a tripartite graph partitioning problem. A normalized cut (NCut) cost function is defined over the partitions. We show that partitioning a tripartite graph is a quadratically constrained quadratic programming (QCQP) problem. For (b), we introduce a new method for incorporating prior knowledge, such as prior community membership, into the current discovery process. The discovery of temporal communities is then performed by threading communities discovered at consecutive time periods using the output from the previous period as prior knowledge. At

each time period, the constrained graph partitioning method is able to capture both the current graph topology and historical information regarding the vertex membership. This problem is efficiently solved using a proposed fractional orthogonal iteration algorithm (instead of pursuing the semidefinite program (SDP) as in [3], which is computationally intractable). We evaluate the proposed approach on synthetic datasets with various settings in order to explore the properties of the new algorithm. A great improvement in clustering precision is observed. In addition, we show the results of applying this method to a sample dataset obtained from CiteSeer (http://citeseer.ist.psu.edu).

2 Problem Statement

This paper considers social networks of researchers in the context of their collaborations on published work. The data in focus includes the co-occurrences of authors with documents, documents with words, and documents with venues. All data are associated with time stamps, which are the years of publication. The data is collapsed on documents yielding the (1) author-word co-occurrences and (2) word-venue co-occurrences, over a certain amount of time. Thus, within each time period there are two correlated bipartite graphs, one associating the authors with the words, and the other associating the words with the venues, which share the same set of vertices of words. We refer to them as a *bipartite graph couple*, which can be seen as a generalized social network of authors, words, and documents. Fig. 2 illustrates two communities in such a social network.

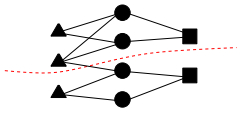

Figure 1. A static social network. Triangles denote the authors, circles denote the words, and rectangles denote the venues. Two static communities are separated by the dashed line.

Over the entire time period, the underlying social network structure is dynamic. Accordingly, instead of observing a single static social network over the entire data set, a sequence of static social networks of various structures is generated, with consecutive snapshots showing significant overlap of entities. A dynamic social network is illustrated in Fig. 2. Three snapshots are included, each having different network structures. It can be seen that each static social network is a bipartite graph couple. The goal of this paper is to cluster authors, words and venues given their changing relationships over time. The desired number of communities k is assumed and given as a parameter.

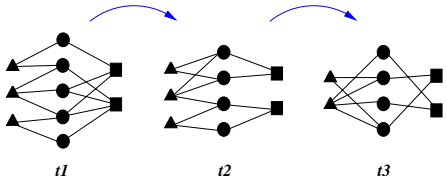

Figure 2. A dynamic social network. Three snapshots are included in the network with various numbers of authors (denoted by triangles), venues (denoted by rectangles), and words (denoted by circles).

3 Community Partitioning

We start from the discovery of static communities from a static social network. Suppose there are two bipartite graphs, $G_{XY} = G(V_X, V_Y, W_{XY})$ and $G_{YZ} = G(V_Y, V_Z, W_{YZ})$, where V_X is the author set, V_Y is the word set, and V_Z is the venue set; $W_{XY} \in \mathbb{R}^{+n_X \times n_Y}$ is the adjacency matrix of graph G_{XY} and $W_{YZ} \in \mathbb{R}^{+n_Y \times n_Z}$ is the adjacency matrix of graph G_{YZ}; n_X, n_Y, n_Z are the size of V_X, V_Y, V_Z.

Now define several indicator matrices: $\hat{X} = [X_1, ..., X_k]$, where X_i is an indicator vector of whether the corresponding element belongs to community i, with 1 indicating so or 0 otherwise. Similarly, we have $\hat{Y} = [Y_1, ..., Y_k]$ and $\hat{Z} = [Z_1, ..., Z_k]$. Given k as the desired number of communities, the cost function of *Normalized Cut (NC)* on a *single* bipartite graph is defined as [7]:

$$J_2 = k - \sum_{i=1}^{k} \hat{X}_i^T D_{XY}^{-\frac{1}{2}} W_{XY} D_{YX}^{-\frac{1}{2}} \hat{Y}_i. \quad (1)$$

where D_{XY} and D_{YZ} are diagonal matrices where the elements are the sums of rows in W_{XY} and W_{YZ}. Meanwhile, in order to obtain a solution efficiently, prior work [7] made two more assumptions on the minimizers: (1) $\hat{X}$, $\hat{Y}$ take real values instead of the discrete set $\{0, 1\}$; (2) $\hat{X}$, $\hat{Y}$ are orthonormal, i.e. $\hat{X}^T\hat{X} = \mathbf{I}$, $\hat{Y}^T\hat{Y} = \mathbf{I}$, $\mathbf{I}$ is an identity matrix.

Now we generalize the cost function for a bipartite graph couple, where we have an additional set of vertices V_Z and the edge weights with V_Y in W_{YZ}.Let J_{XY} be the cost function for G_{XY} and J_{YZ} for G_{YZ}. We introduce a parameter λ to balance the costs on both graphs. Based on Eq. 1, we arrive at a new maximization problem using the new cost function $J_3 = \lambda J_{XY} + (1-\lambda) J_{YZ}$:

$$\begin{aligned} \min_{\hat{X},\hat{Y},\hat{Z}} J_3 &= \min_{\hat{X},\hat{Y},\hat{Z}} (\lambda J_{XY} + (1-\lambda) J_{YZ}) \\ &\equiv \max_{\hat{X},\hat{Y},\hat{Z}} \lambda \sum_{i=1}^{k} \hat{X}_i^T D_{XY}^{-\frac{1}{2}} W_{XY} D_{YX}^{-\frac{1}{2}} \hat{Y}_i \\ &\quad + (1-\lambda) \sum_{i=1}^{k} \hat{Y}_i^T D_{YZ}^{-\frac{1}{2}} W_{YZ} D_{ZY}^{-\frac{1}{2}} \hat{Z}_i \end{aligned} \quad (2)$$

where we again assume that $\hat{X}$, $\hat{Y}$, $\hat{Z}$ are orthonormal, i.e. $\hat{X}^T\hat{X} = \mathbf{I}, \hat{Y}^T\hat{Y} = \mathbf{I}, \hat{Z}^T\hat{Z} = \mathbf{I}$.

Now let us rewrite the problem in matrix form. Define $\widehat{W}_{XY} = D_{XY}^{-\frac{1}{2}} W_{XY} D_{YX}^{-\frac{1}{2}}$ and $\widehat{W}_{YZ} = D_{YZ}^{-\frac{1}{2}} W_{YZ} D_{ZY}^{-\frac{1}{2}}$. Define $U = [U_1, ..., U_k]$, where $U_i = [\hat{X}_i^T, \hat{Y}_i^T, \hat{Z}_i^T]^T$; Let there be a matrix M such that:

$$M = \begin{bmatrix} 0 & \lambda\widehat{W}_{XY} & 0 \\ \lambda\widehat{W}_{XY}^T & 0 & (1-\lambda)\widehat{W}_{YZ} \\ 0 & (1-\lambda)\widehat{W}_{YZ}^T & 0 \end{bmatrix}. \quad (3)$$

Thus, we minimize the matrix trace: $\max_U \text{tr}(U^T M U)$, where $U = [\hat{X}^T, \hat{Y}^T, \hat{Z}^T]^T$. Note there are orthonormal constraints on the segments of U (i.e. $\hat{X}, \hat{Y}, \hat{Z}$).

4 Partitioning Temporal Graphs

Next, we present a constrained graph partitioning method that threads community discovery across consecutive time periods, based on the graph partitioning formulated above.

Graphs with consistent vertices: We first focus on the case where graphs have consistent vertices. For each time period, we have M^t and U^t, where $t = 1, ..., T$ are the time stamps and U^t contains the community membership of authors, words, and venues. Assume that the graphs have consistent vertices; thus, all U^t have the same dimensions. We define a cost function on the difference between two consecutive U's and seek to minimize such difference to achieve "smoothness" in discovery. In particular, we want to $\min_{U^t} c(U^{t-1}, U^t)$ (We refer to U^{t-1} as the reference subspace). In this paper, equivalently, we maximize the square of cosine distances between the U^t and U^{t-1}. Suppose $\dot{X}$, $\dot{Y}$, and $\dot{Z}$ are the reference subspaces for X, Y, Z. We seek to maximize the following:

$$= \text{tr}(U^T \dot{U}\dot{U}^T U) \quad (4)$$

$$= \alpha.\text{tr}(\hat{X}^T \dot{X}\dot{X}^T \hat{X}) + \beta.\text{tr}(\hat{Y}^T \dot{Y}\dot{Y}^T \hat{Y}) + \gamma.\text{tr}(\hat{Z}^T \dot{Z}\dot{Z}^T \hat{Z}) \quad (5)$$

$$= \alpha\|\dot{X}^T \hat{X}\|^2 + \beta\|\dot{Y}^T \hat{Y}\|^2 + \gamma\|\dot{Z}^T \hat{Z}\|^2, \quad (6)$$

where $\dot{U} = [\sqrt{\alpha}\dot{X}^T, \sqrt{\beta}\dot{Y}^T, \sqrt{\gamma}\dot{Z}^T]^T$, α, β and γ are the weight parameters of the membership differences in authors, words, and venues. Notice that $\dot{U}\dot{U}^T$ is essentially the covariance matrix between the vertices in the reference. Since we have assumed consistent vertices in the graphs across different time periods, we essentially minimize the conflicts between the discovered U and the referenced covariance.

Graphs with evolving vertices: Now we generalize the previous section to graphs with evolving vertices. In practice, some vertices may disappear and other new ones may show up, thus the $\dot{U}$ obtained from previous period can disagree with the dimensionality of the U in the current time period. We introduce an additional step to adapt $\dot{U}$ to address this issue.

First, for vertices disappearing from previous time period, since each vertex corresponds to a row in $\dot{U}$, we delete these rows from $\dot{U}$, forming a matrix with the same number of columns but a smaller number of rows, $\dot{U}'$, namely, in the first step shrink():

$$\dot{U}' = \text{shrink}(\dot{U}) = [\dot{X}'^T, \dot{Y}'^T, \dot{Z}'^T]^T. \quad (7)$$

Second, for those new vertices, with no prior knowledge regarding their membership, we require zero co-variances between them and others, resulting in zeros in the corresponding rows. Name this second step expand():

$$\dot{U}'' = \text{expand}(\dot{U}') = [\dot{X}'^T, 0, \dot{Y}'^T, 0, \dot{Z}'^T, 0]^T, \quad (8)$$

where $[\dot{X}'^T, 0]^T$, $[\dot{Y}'^T, 0]^T$, and $[\dot{Z}'^T, 0]^T$ respectively correspond to the newly observed X^t, Y^t, and Y^t; all $0's$ has the appropriate number of rows and k columns.

The two additional steps give raise to the new reference covariance matrix, say $\dot{C} = \dot{U}''\dot{U}''^T$. Incorporate $\dot{C}$ into the original optimization. We arrive at the final formulation of the community discovery problem at each time period:

$$\max_U \text{tr}(U^T M U) + \text{tr}(U^T \dot{C} U) = \max_U \text{tr}(U^T (M + \dot{C}) U) \quad (9)$$

subject to

$$U = [\hat{X}^T, \hat{Y}^T, \hat{Z}^T]^T \text{and} \hat{X}^T, \hat{Y}^T, \hat{Z}^T \text{are orthonormal.} \quad (10)$$

where $M, \dot{C}$ are dependent on the parameters $\lambda, \alpha, \beta, \gamma$, as given in the above.

4.1 Efficient approximate solutions

This section gives an efficient algorithm to solve Eq. 9. It can be seen that Eq. 9-Eq. 10 is quadratically constrained quadratic programming (QCQP) problem, which could have a standard solution by semidefinite programming (SDP) [1]. For example, a related work [3] studied the binary clustering case and proposed an approximate solution using an interior-point method. However, we note that our optimizer here is a matrix ($U = [X^T, Y^T, Z^T]^T$). One might construct a very high-dimensional vector by columns of U and translate the problem into SDP, but difficulty arises from the exploding dimensionality of the problem. Recall that $U \in \mathbb{R}^{(n_X+n_Y+n_Z)\times k}$, where n_X, n_Y, and n_Z are the numbers of authors, words, and venues. The translated SDP problem will have a $k(n_X + n_Y + n_Z)$-dimensional vector as the minimizer (with a $k(n_X + n_Y + n_Z) \times k(n_X + n_Y + n_Z)$ semidefinite matrix of constraints), which can easily surpass the capacity of most SDP solvers.

Instead, we propose an efficient algorithm that searches for approximate solutions, based on algorithms for eigenvectors. First we are aware that the Eq. 9, with orthonormal constraint on U, reaches the maximum when U contains the first k eigenvectors of the symmetric matrix $A = M + \dot{U}\dot{U}^T$. This is a standard result from matrix theory [4]. Second, we seek to preserve the constraints as much as possible while performing the optimization. We modify the *orthogonal iteration* method which is used to calculate the eigenvector space without constraints, arriving at a new method *fractional orthogonal iteration*, presented in Algo. 1.

Here $\text{eig}(A, k)$ calculates the k-dimensional eigenvector space of A without constraints. This is the initial value for the subsequent orthogonal iteration. In the algorithm, step 9 - step 11 produce the normalized $\hat{X}$, $\hat{Y}$ and $\hat{Z}$ as specified in the constraints. Step 8 performs the power iteration as in the original *orthogonal iteration* method for calculating eigenvectors. Up to step 15, the algorithm has projected the original bipartite graph couple into an approximate k-dimensional eigenspace. Then we run k-means to cluster the heterogeneous objects as current communities.

Algorithm 1 fractional orthogonal iteration

$\dot{U} = [\sqrt{\alpha}\dot{X}^T, \sqrt{\beta}\dot{Y}^T, \sqrt{\gamma}\dot{Z}^T]^T$;
$\dot{U}' \leftarrow$ shrink$(\dot{U})$ as in Eq. 7
$\dot{U}'' \leftarrow$ expend$(\dot{U}')$ as in Eq. 8
$\dot{C} \leftarrow \dot{U}''\dot{U}''^T$
$A = M + \dot{C}$
$[U, D] \leftarrow$ eig(A, k)
for $i = 1, 2, 3, \ldots$ **do**
$\begin{bmatrix} \hat{X} \\ \hat{Y} \\ \hat{Z} \end{bmatrix} \leftarrow A \times U$
$Q_X R_X \leftarrow \hat{X}$ // QR factorization
$Q_Y R_Y \leftarrow \hat{Y}$ // QR factorization
$Q_Z R_Z \leftarrow \hat{Z}$ // QR factorization
$U \leftarrow \begin{bmatrix} Q_X \\ Q_Y \\ Q_Z \end{bmatrix}$
end for
$U \leftarrow M \times U$
run k-means on U to obtain the desired partitioning, where each row in U denotes the original data object of the same index.

5 Experiments

A synthetic data generator was created to test the proposed method in various conditions, including different edge density-to-noise ratio, various proportions of $X/Y/Z$, different settings of λ, and different numbers of clusters (k). Two connected graphs G_{XY} and G_{YZ} are generated for the prescribed K and sizes of X, Y, and Z. All clusters contain the same number of entities with specified proportions of $X, Y, andZ$. The densities of all the clusters are the same, but the edge weights vary randomly. Random noise is added to the graph and density is determined by the given noise-signal ratio parameter (nsr). Setting $nsr = 1$ yields a random graph without cluster structures. Presumably, the community structures in the graph XY diminish as the noise-signal ratio (nsr) grows. Low nsr indicates that graph partitioning will be easier. The table below includes a complete list of parameters and their meanings.

abbr.	usages
fsi	fractional subspace iteration
par	partitioning static graphs using fsi
t-par	partitioning temporal graphs using fsi
k	number of clusters
$density$	the edge density of the graph clusters
nsr	noise-signal ratio, noise density / cluster density
x/z	the size of X / the size of Z
λ	the weight parameter in Eq. 3

5.1 Precision w.r.t. graph conditions

We perform fsi on different settings of x/z ratios for a fixed setting of λ. In real world datasets, the sizes X and Z are usually not balanced. We compare fsi with *subspace iteration* for imbalanced data against fsi by varying the x/z ratio. Fig. 3 shows different settings of x/z for different densities. Recall that a large x/z indicates that the size of X is much greater than that of Z. Without loss of generality, we assume $x/z \geq 1$. We can see that for sparse graphs (small density) the fsi outperforms *subspace iteration* greatly (illustrated in the subfigure on the bottom). In simple cases (large density), the fsi generally outperforms *subspace iteration* for small x/z; however, fsi under-performs *subspace iteration* slightly for small x/z on dense graphs. Note that real-world graphs are usually very sparse; thus, fsi could be favored on many real-world datasets.

5.2 Precision w.r.t. parameter settings

Here we test different settings of parameters and their impact on community discovery precision. A set of experiments were run with different settings of λ in different x/z ratios. The results illustrated in Fig. 4 show that the favorable λ are different when x/z varies. When the X outnumbers Z by a large margin, a greater value in λ is favored; similarly, small λ performs better when there are few X entities compared with Z. This suggests that graphs with more edges deserve a larger weight in the cost evaluation.

Finally, we compare fsi with *subspace iteration* on different numbers of clusters, at different *subspace iteration*. We can see that, for large density, fsi still outperforms *subspace iteration* for large numbers of clusters. However, the *subspace iteration* seems to work better than fsi for the case of many clusters on sparse graphs. In practice, we can substitute fsi by recursively performing k-means using $k = 2$ for bi-partitioning the graph, similar to [7].

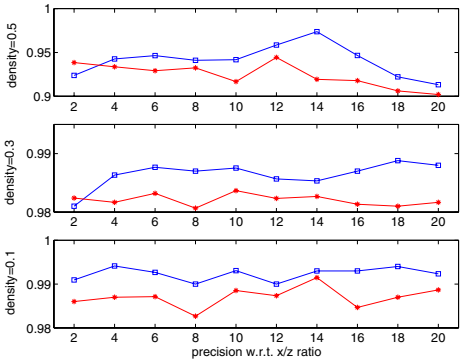

Figure 3. The precision w.r.t. different x/z ratio, at different edge density levels. Here $nsr = 0.1$, $k = 2$, $\lambda = 0.5$.

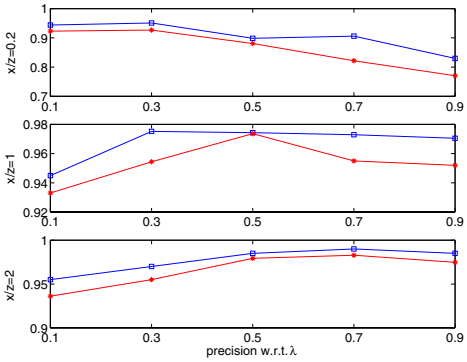

Figure 4. The precision w.r.t. λ, at different x/z ratio. Here $density = 0.3$, $nsr = 0.3$, $k = 2$.

Venues	1969-94	1994-96	1996-98	1998-2000	2000-02	2002-04
JMLR	M I Jordan	M I Jordan	W L Johnson	S Thrun	D Koller	A Blum
	L P Kaelbling	L P Kaelbling	N Friedman	C Boutilier	A W Moore	S Thrun
	J Y Halpern	Z Ghahramani	D Koller	T Sandholm	M I Jordan	S Zilberstein
PAMI	S P Singh	S P Singh	R E Schapire	D Koller	M L Littman	P Stone
	Z Ghahramani	M K Warmuth	Y Singer	N Friedman	S Thrun	J Langford
	M K Warmuth	T G Dietterich	R Dechter	Y Singer	D Schuurmans	T Eiter
ICML	T G Dietterich	T Dean	T J Sejnowski	A Mccallum	J Shawe-taylor	P Domingos
	T Dean	Y Bengio	H S Seung	L P Kaelbling	S P Singh	A K Jain
	Y Bengio	P Smets	D Poole	S P Singh	N Friedman	S Baker
AAAI/IAAI	P Smets	W Maass	M I Jordan	P R Cohen	N Cristianini	S Chawla
	W Maass	V Tresp	N Tishby	R Khardon	A Mccallum	R Dechter
	V Tresp	D Weinshall	R Greiner	M J Kearns	P Domingos	C Guestrin
UAI	D Weinshall	D Geiger	Y Mansour	K Nigam	Y Bengio	C Boutilier
	D Geiger	S Kambhampati	M K Warmuth	N Cristianini	D Freitag	M J Kearns
	D Poole	A Saffiotti	Y Freund	J Shawe-taylor	A Y Ng	T Lukasiewicz
IJCAI	R E Schapire	R E Schapire	D P Helmbold	C Baral	M K Warmuth	A Demiriz
	S Kambhampati	D S Nau	C Boutilier	A W Moore	G E Hinton	S P Singh
	C Baumlckstroumlm	H A Simon	M L Littman	D Fox	N Tishby	D Koller
JAIR	F Bacchus	F Bacchus	P Dayan	D Roth	A J Smola	D Schuurmans
	A Saffiotti	D Poole	A J Grove	M P Wellman	G Raumltsch	S Prabhakar

Table 1. Machine learning community during 1969-2004 in a CiteSeer sample.

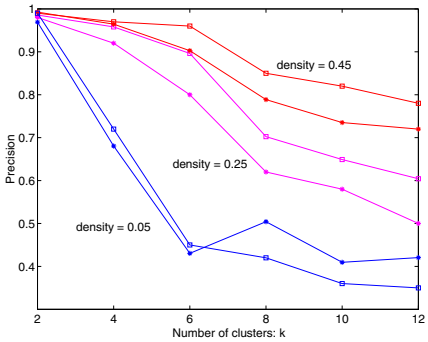

Figure 5. The precision w.r.t. k, at different densities: $density = 0.05$**,** $density = 0.25$**,** $density = 0.45$**.**

5.3 Higher precision by prior knowledge

The fsi algorithm uses the discovery results from the previous time period as prior knowledge for analyzing temporal graphs. This knowledge is then used as an additional constraint while discovering communities in the current time period. We simulate a 2-period temporal graph where communities in the first time period are clearly defined and then the community structure becomes vague in the second time period. The community membership from the first time period is used as the prior knowledge in the second time period. (In practice, we can also allow manual manipulation of entity membership for the *first time period*, in order to increase the validity of this assumption.)

methods	precisions	methods	precisions
par on g_1	0.9193	par on g_{12}	0.8212
par on g_2	0.2123	t-par on g_{12}	0.9169
average of the above	0.5658		

Table 2. Different methods on temporal graphs.

In Table 2, we illustrate the precisions of clustering on the snapshots from each time period and the average precision. It can be seen that the static partitioning precision is very high on g_1 (0.9193) and very low on g_2 (0.2123): the average of the two is about 0.5658. In addition, we perform clustering on the graph over the complete time periods, obtaining a precision of 0.8212. Then we perform the constrained partitioning t-par on the temporal graph, yielding the precision 0.9169. The precision is much higher than performing clustering periodically or on the complete graph.

5.4 Real-world dataset and experiments

A real-world data set for further experimentation was generated by sampling documents from CiteSeer using combined document metadata from CiteSeer, 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. A set of venues was chosen from five fields in computer science (software engineering, data mining, artificial intelligence, databases, and distributed computing), such that data from each field included at least 2000 distinct author names and at least ten years of significant coverage. All documents contained in CiteSeer from each venue were obtained and the top 100 keyphrases were extracted from each document using the KEA keyphrase extraction algorithm (http://www.nzdl.org/Kea/). The final dataset contained 12,677 authors and 45,295 keyphrases from 30 distinct venues ranging over the years 1969 to 2004. The total number of documents used was 13,310.

Experiments on this data set began by empirically determining the appropriate number of clusters. While it is an open problem to determine the dimension of a subspace for embedding a graph, we used simple heuristics. We ran the proposed community discovery algorithm (fsi) with different k and chose the k corresponding to the smallest $\widetilde{J}$ (or the greatest $\gamma = \text{tr}(U^T(M + \dot{C})U)$) as in Eq. 9. We observed that the γ initially grows dramatically as k increases, but grows at a much lower rate as k becomes large. Thus

Venues	1969-94	1994-96	1996-98	1998-2000	2000-02	2002-04
PODS	M Yannakakis	M Yannakakis	R Hull	A Mendelson	G Gottlob	S Abiteboul
	V Vianu	V Vianu	A Mendelzon	J Paredaens	V Vianu	L Popa
	A Gupta	J Y Halpern	Z M Zsoyoglu	C Papadimitriou	H Garcia-molina	T Milo
	Garciacute	Garciacute	H Garcia-molina	H Garcia-molina	J Widom	P G Kolaitis
SIGMOD	J Widom	J Widom	D Suciu	S Abiteboul	A Y Halevy	P S Yu
	J F Naughton	H Garcia-molina	A Silberschatz	D Florescu	C Faloutsos	F Neven
	H Garcia-molina	J F Naughton	A Y Levy	A Y Levy	D Suciu	C Beeri
	C Faloutsos	C Faloutsos	L Libkin	R Motwani	D Gunopulos	R Rastogi
VLDB	A Kemper	J Hammer	G Moerkotte	L V S Lakshmanan	S Lee	J Han
	K Ramamritham	A Biliris	S Seshadri	T Milo	J Han	D Srivastava
	G Moerkotte	K Ramamritham	S Abiteboul	S Cluet	W Fan	M N Garofalakis
	I S Mumick	A Kemper	J Widom	J Han	R Rastogi	J Widom
SIGMOD Record	A Biliris	C Baumlckstroumlm	R Agrawal	D Suciu	C S Jensen	A Y Halevy
	J Hammer	G Moerkotte	R Ramakrishnan	J S Vitter	H V Jagadish	C Li
	M Chen	I S Mumick	S Sudarshan	R Rastogi	D Kossmann	J Madhavan
	P S Yu	K Lin	K Ramamritham	G D Giacomo	D Srivastava	W Fan
ICDM	T Milo	S Berson	A Kemper	C S Jensen	K Chakrabarti	B Babcock
	D Suciu	D Suciu	D Florescu	D Srivastava	S Muthukrishnan	C Y Chan
	J Han	D Kossmann	P Atzeni	O Shehory	D S Weld	C Koch
	K Lin	C A Knoblock	M Benedikt	M Lenzerini	G D Giacomo	J Gehrke

Table 3. Database community during 1969-2004 in a CiteSeer sample.

we chose the smallest k that gave the near maximum γ. This gave us $k = 4$.

Then we ran the temporal community discovery (t-par) algorithm with $k = 4$ with various settings of λ. For screening the results, we judge the quality of discovery by examining the grouping of venues since their number is small. We observed that the quality is better for greater λ, supporting the results from synthetic datasets that suggest λ should be set proportionally to $|X|/|Z|$. Here we set $\lambda = 0.6$.

We observe that the resulting communities of authors, venues, and words are well grouped. Four communities are discovered for *artificial intelligence and machine learning*, *database and data mining*, *parallel and distributed computing*, and *software engineering*. We present two discovered communities and their authors in Table 1 and Table 3. In our experiments, we used the discovered venue set to manually produce community labels. The keyphrases (ranked by frequency) were considered as the summarization of a community.

Table 1 includes a subset of authors discovered in the *artificial intelligence and machine learning* community over six time periods. For presentation, we rank the authors by their number of papers within the corresponding periods. We can observe that the community memberships of authors are relatively stable but change over time. In the experiments, we observed that the top authors remained as the "core" members of the corresponding community and there were many more authors who had joined and left from the communities during these six time periods. The leftmost column shows the top venues. Similarly, authors from the *database and data mining* community are presented in Table 3.

6 Conclusion

This paper addresses an emerging problem of temporal community discovery from communication documents, by which one can observe the temporal trends in community membership over time. The problem is formulated as a tripartite graph partitioning problem with prior knowledge available of entity covariances. Temporal communities are discovered by threading the partitioning of graphs in different time periods, using a new constrained partitioning algorithm. Evaluation of the new algorithm is carried out on several synthetic datasets and a real-world dataset prepared from CiteSeer. Experiments on synthetic data reveal the properties of the new algorithm in various graph conditions. Experiments on CiteSeer data show the effectiveness of the proposed approach in author community discovery. Future work will seek to track the community membership of individuals over time and investigate the applicability of the proposed methods to different domains such as viral marketing or recommendation services. Additionally, temporal topical trends will be further investigated.

References

[1] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[2] C. Ding. A tutorial on spectral clustering. In *Proc. of the 25th International Conference on Machine Learning*, July 2004.

[3] B. Gao, T.-Y. Liu, X. Zheng, Q.-S. Cheng, and W.-Y. Ma. Consistent bipartite graph co-partitioning for star-structured high-order heterogeneous data co-clustering. In *KDD '05: Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 41–50, New York, NY, USA, 2005. ACM Press.

[4] G. H. Golub and C. F. V. Loan. *Matrix Computations*. The Johns Hopkins University Press, 1996.

[5] D. Harel and Y. Koren. Clustering spatial data using random walks. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, 2001.

[6] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth. The author-topic model for authors and documents. In *UAI '04: Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 487–494. UAI Press, 2004.

[7] H. Zha, X. He, C. Ding, H. Simon, and M. Gu. Bipartite graph partitioning and data clustering. In *CIKM '01: Proceedings of the tenth international conference on Information and knowledge management*, pages 25–32, New York, NY, USA, 2001. ACM Press.

[8] 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.