The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

CoDiNMF: Co-Clustering of Directed Graphs via NMF

Woosang Lim
School of Computational Science
and Engineering
Georgia Institute of Technology,
Atlanta, GA 30332, USA
woosang.lim@cc.gatech.edu

Rundong Du
School of Computational Science
and Engineering
Georgia Institute of Technology,
Atlanta, GA 30332, USA
rdu@gatech.edu

Haesun Park
School of Computational Science
and Engineering
Georgia Institute of Technology,
Atlanta, GA 30332, USA
hpark@cc.gatech.edu

Abstract

Co-clustering computes clusters of data items and the related features concurrently, and it has been used in many applications such as community detection, product recommendation, computer vision, and pricing optimization. In this paper, we propose a new co-clustering method, called CoDiNMF, which improves the clustering quality and finds directional patterns among co-clusters by using multiple directed and undirected graphs. We design the objective function of co-clustering by using min-cut criterion combined with an additional term which controls the sum of net directional flow between different co-clusters. In addition, we show that a variant of Nonnegative Matrix Factorization (NMF) can solve the proposed objective function effectively. We run experiments on the US patents and BlogCatalog data sets whose ground truth have been known, and show that CoDiNMF improves clustering results compared to other co-clustering methods in terms of average F1 score, Rand index, and adjusted Rand index (ARI). Finally, we compare CoDiNMF and other co-clustering methods on the Wikipedia data set of philosophers, and we can find meaningful directional flow of influence among co-clusters of philosophers.

Introduction

Clustering is an essential task in unsupervised learning which finds the inherent relationships and group structures in the data sets. In particular, co-clustering simultaneously computes co-clusters which consist of both features and data items or higher order entities at the same time by exploiting the dualities. Since co-clustering is effective in analyzing the dyadic or higher-order relationships compared to the traditional one-way clustering, it has many applications such as text mining (Dhillon 2001; Dhillon, Mallela, and Modha 2003), bioinformatics (Cho et al. 2004), product recommendation (Vlachos et al. 2014), pricing optimization (Zhu, Yang, and He 2015), sense discovery (Chen et al. 2015), and community detection (Rohe, Qin, and Yu 2016).

Dhillon (Dhillon 2001) suggested Spectral Co-clustering (SCo) to compute co-clusters by using the min-cut problem of a bipartite graph with documents and words as parts. Since then, co-clustering has been studied with different ways over the last decades; information theoretic co-clustering (Dhillon, Mallela, and Modha 2003), multi-view co-clustering (Sun et al. 2015), co-clustering based on spectral approaches (Wu, Benson, and Gleich 2016; Rohe, Qin, and Yu 2016), co-clustering based on NMF (Long, Zhang, and Yu 2005; Wang et al. 2011). However, most of co-clustering methods assume that the connections between entities are symmetric or undirected, but many interactions in real networks are asymmetric or directed. For example, the data set of patents and words contains a citation network among patents, and it can be represented as a directed and asymmetric graph. For development of philosophy, there are influence flow from one philosopher or philosophy school to other philosophers or schools respectively, and these relationships can be directed and asymmetric. Rohe et al. recently proposed a spectral co-clustering method for directed networks, called DI-SIM, which uses the asymmetric regularized graph Laplacian for directed graph (Rohe, Qin, and Yu 2016). DI-SIM computes the left and right singular vectors of regularized graph Laplacian to generate two lower-dimensional representations for sending and receiving nodes. Then, by using an asymmetry score, it discovers the asymmetries in the relationships and describe the directional communities. However, DI-SIM can only be applied to the data sets which consist of one kind of entity distinguished by sending and receiving roles.

In this paper, we propose a new NMF-based co-clustering method, called CoDiNMF, which computes co-clusters by using both directed and undirected relationships to improve co-clustering quality. Especially, CoDiNMF is able to find directional relationships among co-clusters by controlling the net directional flow between different co-clusters. In the later sections, we will derive CoDiNMF, and compare it with other NMF-based co-clustering in terms of accuracy on the US patents and BlogCatalog data sets. We will also demonstrate its ability for finding directional relationships on the Wikipedia data set of philosophers.

Related Work

In this section, we briefly discuss some of the existing co-clustering methods in literature including spectral co-clustering, and NMF based co-clustering.

Spectral Co-clustering

Let X be the $m \times n$ data matrix for the set of features $A = \{a_1, ..., a_m\}$ and the set of data items $B = \{b_1, ..., b_n\}$. For a document data set, A is the set of words, B is the set of documents, and X is the weight matrix between A and B. Dhillon suggested Spectral Co-clustering (SCo) which constructs graph Laplacian by using a bipartite graph represented by the edge weight matrix $M = \begin{bmatrix} 0 & X \\ X^\top & 0 \end{bmatrix}$ to computes low dimensional embedding, and it runs k-means clustering on the computed embedding to find k co-clusters (Dhillon 2001). More specifically, the objective function of co-clustering for bipartition is designed to solve min-cut problem defined with a partition vector q and the weight matrix M, where $q_i = \sqrt{\frac{\eta_2}{\eta_1}}$ if the i-th row/column of M belongs to the first cluster and $q_i = -\sqrt{\frac{\eta_1}{\eta_2}}$ if the i-th row/column of M belongs to the second cluster, η_i is the sum of edge weights of the nodes in the i-th cluster, and the plus or minus sign of q_i plays an important role in assigning nodes to clusters. Since the problem of solving the proposed objective function is NP-complete, SCo was suggested. It computes the second left and right singular vectors of normalized weight matrix $X_n = D_1^{-1/2} X D_2^{-1/2}$, and construct vector $y \in \mathbb{R}^{m+n}$ as an approximation of q, where D_1 and D_2 are diagonal matrices such that $D_1(i,i) = \sum_j X_{i,j}$, and $D_2(i,i) = \sum_j X_{j,i}$. Finally, it runs k-means algorithm on the 1-dimensional y to compute co-clusters. For multipartitioning, it uses $\ell = \lceil \log_2 k \rceil$ singular vectors of X_n to construct a matrix $Y \in \mathbb{R}^{(m+n) \times \ell}$ and runs k-means on Y.

Wu et al. extended the spectral co-clustering method for the tensor data sets and proposed a method called General Tensor Spectral Co-clustering (GTSC) (Wu, Benson, and Gleich 2016). First, GTSC extends the original rectangle data tensor to a square and symmetric extended tensor. Next, it constructs a Markov chain by using the transition matrix and the super-spacey stationary vector, and computes the second leading real-valued eigenvector of Markov chain. At the final step, it recursively applies sweep cut for the recursive bisection procedures for clustering. Although GTSC extended spectral co-clustering, it still uses undirected and symmetric connections between entities.

Rohe et al. recently proposed a new spectral co-clustering method called DI-SIM, which applies the spectral co-clustering approach to directed graph (Rohe, Qin, and Yu 2016). First, it assumes that we have an $n \times n$ weight matrix S which represents directed connections among n nodes in the data set, and it constructs a regularized graph Laplacian by normalizing S as $L = O^{-1/2} S P^{1/2}$, where O, P are $n \times n$ diagonal matrices with $O_{i,i} = \sum_j S_{i,j} + \tau$ and $P_{i,i} = \sum_j S_{j,i} + \tau$, and the regularization parameter τ is the average out-degree. Next, it computes the first k left and right singular vectors of L, and constructs a k-dimensional embedding by normalizing rows of singular vectors. At the final step, it runs k-means on the k-dimensional embedding and finds the asymmetry in the relationships by using an asymmetry score. Although DI-SIM can analyze the clustering asymmetries on some data sets, it can be applied only to the special case of data set which represents sending and receiving relationships among the nodes of the same kind of entities.

NMF-based Co-clustering

Non-negative Matrix Factorization (NMF) and its variants have been used for clustering (Xu, Liu, and Gong 2003; Kim and Park 2011; Kuang, Yun, and Park 2015; Kuang and Park 2013; Du et al. 2017a), and NMF has the advantage of providing intuitive interpretation for the result. Non-negative Block Value Decomposition (NBVD) was proposed to analyze the latent block structure of the non-negative data matrix based on Nonnegative Matrix Tri-Factorization (NMTF) (Long, Zhang, and Yu 2005), and its objective function is

$$\underset{W,T,H}{\text{minimize}} \; \|X - WTH\|_F^2 \;\; \text{subject to} \;\; W, T, H \geq 0, \qquad (1)$$

where X is the data matrix defined in the previous section, W is $m \times c$, T is $c \times l$, and H is $l \times n$. T is called as block value matrix which can be considered as a compact representation of X. W and H are coefficient matrices for row and column clusters, respectively. To solve Eqn (1), they derived an algorithm that iteratively updates the decomposition by using a set of multiplicative updating rules.

Wang et al., proposed Locality Preserved Fast Nonnegative Matrix Tri-Factorization (LP-FNMTF) for co-clustering which uses two undirected graphs G_A of features and G_B of data points in addition to the bipartite graph of X (Wang et al. 2011), where the undirected graphs G_A and G_B can be either generated from X or obtained from prior knowledge (Wang et al. 2011). Thus, the objective function of LP-FNMTF in Eqn (2) has two terms in addition to Eqn (1) to consider the manifold structures in both data and feature space, and the constraints in Eqn (1) and Eqn (2) are different.

$$\begin{aligned} &\underset{W,S,H}{\text{minimize}} \; \|X - WSH\|_F^2 \qquad (2) \\ &+\alpha \, \mathrm{tr}(W^T(I - L_A)W) + \beta \, \mathrm{tr}(H(I - L_B)H^T) \\ &\qquad\qquad \text{subject to } W \in \Psi^{m \times d}, H \in \Psi^{c \times n}, \end{aligned}$$

where L_A and L_B are normalized graph Laplacians s.t. $L_A = D_A^{-1/2} G_A D_A^{-1/2}$ and $L_B = D_B^{-1/2} G_B D_B^{-1/2}$, and Ψ is a cluster indicator matrix. Basically, LP-FNMTF computes $d \times c$ co-clusters, but we can set $d = c = k$ for finding k clusters among data points.

So far, we have discussed spectral co-clustering and NMF-based co-clustering including NBVD and LP-FNMTF. All of them, except DI-SIM, use only undirected relationships for constructing the objective function to compute k co-clusters, and DI-SIM only considers the data set consisting of sending and receiving relationships among the nodes of the same kind of entities.

Co-clustering of Directed Graphs

Many data sets in the real world contain both undirected and directed relationships. For example, research papers and patents can be encoded with words as their features, but the

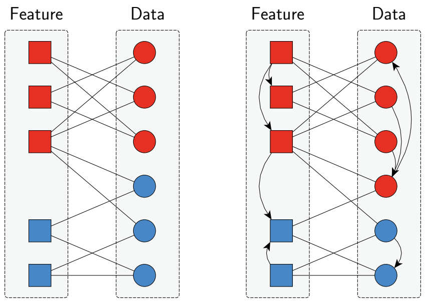

Figure 1: Previous co-clustering model uses undirected bipartite graph (left figure), and our co-clustering model uses bipartite graph combined with directed subgraphs (right figure)

two data sets also have directed relationships as the citations among the papers and the citations among the patents, respectively. Fig 1 describes an example of dyadic data which can be represented as combination of both an undirected subgraph and directed subgraphs. Thus, unlike the clustering methods on the multiple undirected graphs (Tang, Lu, and Dhillon 2009; Wang et al. 2011; Sun et al. 2015) and multi-view graphs on the same entity (Nie, Li, and Li 2017), we consider an extended $(m+n)\times(m+n)$ weight matrix M by combining the weight matrices of undirected graph and directed graphs to encode additional directed relationships for the data set of two entities.

$$M = \begin{bmatrix} Z & X \\ X^\top & S \end{bmatrix} \tag{3}$$

where $X \in \mathbb{R}^{m\times n}$ is a weight matrix of undirected graph, $Z \in \mathbb{R}^{m\times m}$ and $S \in \mathbb{R}^{n\times n}$ can be weight matrices of directed graphs shown on the right side in Fig 1. For a document data set, Z can be a weight matrix of word-word graph, X a weight matrix of word-document graph, S a weight matrix of document-document graph.

More in general, we can assume that relationships between features and items are directed, then X_1 and X_2 in Eqn (4) can be different, *i.e.*, $X_1 \neq X_2$.

$$M = \begin{bmatrix} Z & X_1 \\ {X_2}^\top & S \end{bmatrix}. \tag{4}$$

Since such data inherently have directional relationships among the entities, we need to exploit them to enhance the clustering quality and also to find directional relationships among co-clusters. Our CoDiNMF was derived under the condition of Eqn (4), and we discuss it in detail in the following section.

CoDiNMF: Co-clustering of Directed Graphs via NMF

We first define a directional cut in directed graphs to derive the objective function of CoDiNMF. Let M in Eqn (4) be the extended weight matrix of directed graph, and V_i be a set of i-th co-cluster which consists of both entities of set A and B. Then, the directional cut in directed graph for cluster V_i and V_j is $\mathrm{Cut}(V_i, V_j) = \sum_{s\in V_i, t\in V_j} M_{s,t}$, where $\mathrm{Cut}(V_i, V_j) \neq \mathrm{Cut}(V_j, V_i)$ since the cuts between V_i and V_j can be different depending on directions. Based on the notion of directional cut, we introduce two different aspects which are cut minimization in directed graphs and control of net directional flow to derive an objective function.

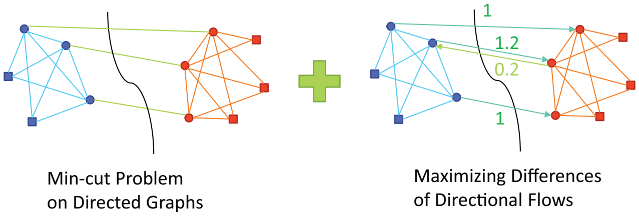

Figure 2: Illustrations of the objective function. Min-cut problem of directed graphs can be derived by considering the sum of all directional cuts between different co-clusters (left figure). An additional term helps to control a net directional flow which is the sum of differences of directional flows (right figure).

Edge cut minimization in directed graphs: Our first criteria is minimizing cut which is a sum of all directional cuts among k co-clusters. We assume that each co-cluster indicator vector u_i consists of 0 and 1 whereas usual spectral clustering based methods consider -1 and 1 for the values of cluster indicator vector. Specifically, the elements of u_i are 1 for the members in the i-th cluster and 0 for the members in the other clusters, and u_i is non-negative and $u_i^\top u_j = 0$ when $i \neq j$. By using the defined u_i, the directional cut from V_i to V_j can be expressed as $\mathrm{Cut}(V_i, V_j) = u_i^\top M u_j$. Then, the cut becomes

$$\sum_{i\neq j} \mathrm{Cut}(V_i, V_j) = \sum_{i<j} (u_i^\top M u_j + u_j^\top M u_i), \tag{5}$$

and Lemma 1 provide its simplified form.

Lemma 1 *Given M in Eqn (4), let $u_i \in \mathbb{R}^{(m+n)\times 1}$ be the $0-1$ co-co-cluster indicator vector of i-th cluster. Then, the sum of directional cuts for k co-clusters is*

$$\frac{1}{2}\left(e^\top (M+M^\top)e - \sum_{i=1}^{k} u_i^\top (M+M^\top) u_i\right), \tag{6}$$

where $e \in \mathbb{R}^{(m+n)\times 1}$ is a vector with all entries one, and M is the weight matrix of directed graphs defined in Eqn (4).

Proof 1 *By using co-cluster indicator vectors u_i and u_j, we have $\mathrm{Cut}(V_i, V_j) = u_i^\top M u_j$. Then, the sum of all directional cuts among k co-clusters is $\sum_{i<j}(u_i^\top M u_j + u_j^\top M u_i)$. Thus, we have $\sum_{i<j}(u_i^\top M u_j + u_j^\top M u_i) = \frac{1}{2}\sum_{i=1}^{k}\sum_{i\neq j}(u_i^\top (M+M^\top) u_j) = \frac{1}{2}(e^\top (M+M^\top)e - \sum_{i=1}^{k} u_i^\top (M+M^\top) u_i)$, where $e = \sum_{i=1}^{k} u_i$.*

Thus, we can use Eqn (6) to minimize cut.

Controlling the effect of the net directional flow: To further incorporate directed relationship, we consider an additional term which controls the sum of net directional flows between different co-clusters as described in Fig 2. where the net directional flow between two nodes can be defined as $|M_{s,t} - M_{t,s}|$. Then, the sum of net directional flows between co-clusters V_i and V_j can be defined as $\sum_{s \in V_i, t \in V_j} |M_{s,t} - M_{t,s}|$, and it has an equivalent form s.t.

$$\sum_{s \in V_i, t \in V_j} |M_{s,t} - M_{t,s}| = u_i^\top |M - M^\top| u_j, \qquad (7)$$

where $|M - M^\top|_{s,t} = |M_{s,t} - M_{t,s}|$. Lemma 2 provides a simplified form of sum of net directional flows, and we can control its amount by using Eqn (8).

Lemma 2 *The sum of net directional flows among k co-clusters is*

$$\frac{1}{2}(e^\top |M - M^\top| e - \sum_{i=1}^{k} u_i^\top |M - M^\top| u_i). \qquad (8)$$

Proof 2 *By using co-cluster indicator vectors u_i and u_j, the sum of net directional flow between two co-clusters V_i and V_j can be expressed as $\sum_{s \in V_i, t \in V_j} |M_{s,t} - M_{t,s}| = u_i^\top |M - M^\top| u_j$, where $|M - M^\top|_{s,t} = |M_{s,t} - M_{t,s}|$. Then, the total sum of net directional flow among k co-clusters becomes $\sum_{i<j}(u_i^\top |M - M^\top| u_j) = \frac{1}{2}\sum_{i=1}^{k}\sum_{i \neq j}(u_i^\top |M - M^\top| u_j) = \frac{1}{2}(e^\top |M - M^\top| e - \sum_{i=1}^{k} u_i^\top |M - M^\top| u_i)$.*

Objective function: If we increase the sum of net directional flows, then we can find more directional patterns among k co-clusters. Thus, we consider the sum of net directional flows (Eqn (5)) as an additional term to the min-cut problem (Eqn (7)), then we have the following objective function

$$\underset{u_1,\ldots,u_k}{\text{minimize}} \sum_{i<j} \left(u_i^\top (M + M^\top) u_j - \lambda u_i^\top |M - M^\top| u_j \right), \qquad (9)$$

where $\lambda \in [0, 1]$ adjusts the effects of the term which controls the sum of net directional flows.

Proposition 1 *The objective function in Eqn (9) can be simplified as follows*

$$\underset{u_1,\ldots,u_k}{\text{maximize}} \sum_{u_i} u_i^\top (M') u_i, \qquad (10)$$

where $M' = (M + M^\top - \lambda |M - M^\top|)$ is nonnegative and symmetric, and u_i is the $0 - 1$ co-cluster indicator vector.

Proof 3 *The alternative forms of $\sum_{i<j} u_i^\top (M + M^\top) u_j$ and $\sum_{i<j} u_i^\top |M - M^\top| u_j$ are Eqn (6) in Lemma 1 and Eqn (8) in Lemma 2, respectively, and the equivalence between Eqn (9) and Eqn (10) can be shown by using them.*

Although we use directional flows and cuts to find the directional patterns, the linear transformed matrix M' in Eqn (10) becomes nonnegative and symmetric, since

$$M' = \begin{bmatrix} Z' & X' \\ X'^\top & S' \end{bmatrix}, \qquad (11)$$

where $Z' = Z + Z^\top - \lambda |Z - Z^\top|$, $S' = S + S^\top - \lambda |S - S^\top|$, $X' = X_1 + X_2 - \lambda |X_1 - X_2{}^\top|$, and $\lambda \in [0, 1]$.

So far, we have assumed the generalized condition that Z and S in Eqn (4) are asymmetric, and $X_1 \neq X_2$. However, if $Z \neq Z^\top$, $S \neq S^\top$, and $X_1 = X_2$ like Eqn (3), then we can only apply λ parameter to the asymmetric weight matrices Z and S, *i.e.*, $M' = \begin{bmatrix} Z' & 2X \\ 2X^\top & S' \end{bmatrix}$, where $X = X_1 = X_2$. The rest of the cases can be similarly defined, too.

CoDiNMF: We show that the objective function of co-clustering (Eqn (10)) can be solved by using NMF with some relaxation. We keep the non-negativity constraints of M' and u_i, and derive an alternative form of Eqn (10) with the temporal constraint of unit norm for u_i. We replace the 1 values in the u_i as $\frac{1}{\sqrt{|V_i|}}$, then components of u_i have $\frac{1}{\sqrt{|V_i|}}$ for the members of the i-th cluster and 0 for the members of the other clusters, and this condition helps to make the size of clusters more balanced. As a result, the constraint of u_i becomes $u_i^\top u_j = \delta_{ij}$ and $u_i \geq 0$, where δ_{ij} is the Kronecker delta. With this condition, we have a modified objective function

$$\underset{u_1,\ldots,u_k}{\text{maximize}} \sum_{u_i} u_i^\top (M') u_i \;\; \text{s.t.} \;\; u_i^\top u_j = \delta_{ij}, \; u_i \geq 0. \qquad (12)$$

Proposition 2 provides an alternative form of Eqn (12) which is related to nonnegative matrix factorization.

Proposition 2 *Suppose that we have the modified objective function as Eqn (12). Then, it is equivalent to Eqn (13)*

$$\underset{U}{\text{minimize}} \, \|M' - UU^T\|_F^2 \;\; \textit{s.t.} \;\; U^\top U = I, \; U \geq 0, \qquad (13)$$

where the k columns of $U \in \mathbb{R}^{(m+n) \times k}$ are the co-cluster indicator vectors.

Proof 4 *The constraint $u_i^\top u_j = \delta_{ij}$ and $u_i \geq 0$ can be represented as $U^\top U = I$ and $U \geq 0$, respectively. Then,* $\text{maximize}_{u_1,\ldots,u_k} \sum_{u_i} u_i^\top (M') u_i$ *is equivalent to* $\text{minimize}_U \, \text{tr}(M'^\top M' - 2M'^\top UU^\top)$ *under the condition of $U^\top U = I$ and $U \geq 0$. Thus, we have Eqn (13).*

We note that there are two approaches to solve the modified objective function. If we discard the non-negativity constraint of u_i, then the spectral approaches can be applied to solve the objective function Eqn (12). Whereas we keep the non-negativity constraint of u_i, but discard the orthonormal condition of u_i to solve the objective function Eqn (13). Then, we can apply SymNMF (Kuang, Yun, and Park 2015) to Eqn (13), and compute the co-clusters by comparing the entries in row vectors of U.

Normalization of directed subgraphs: Before we compute M' in Eqn (13), we have to construct M in Eqn (3), and we need directed graphs Z and S which can be given or generated from X, *e.g.*, k-nearest neighbors. Meanwhile, we can use normalized graph Laplacian of Z and S instead of original Z and S to more effectively represent the inherent structures of directed graphs. For example, we can

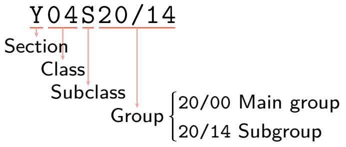

Figure 3: CPC scheme

Table 1: US Patent data sets. There are numbers of patents, citations, and groups for the patent classes.

patent class	# of patents	# of citations	# of groups
A22	4976	28746	230
B06	2938	11549	82
B68	790	2433	93
C14	583	1125	69

normalize S as $L_S = O_S^{-1/2} S P_S^{-1/2}$, where O_S, P_S are $n \times n$ diagonal matrices with $O_S(i,i) = \sum_j S_{i,j} + \tau$ and $P_S(i,i) = \sum_j S_{j,i} + \tau$, and the regularization parameter τ can be 0 or any nonnegative real value. We replace Z and S in Eqn (3) as L_Z and L_S, respectively. Then, we have $M = \begin{bmatrix} L_Z & X \\ X^\top & L_S \end{bmatrix}$. We note that we can use other asymmetric matrices instead of graph Laplacians or original matrices, if they are suitable for finding directional flows.

The sizes and Frobenius norms of L_Z, L_S, and X are usually different, and we can control their contributions by adopting balance parameter α and β as in Eqn (2). Then, we have

$$M = \begin{bmatrix} \alpha L_Z & X \\ X^\top & \beta L_S \end{bmatrix}. \tag{14}$$

For Z and S, the balance parameters α and β can be set as $\alpha = c_1 \|X\|_F / \|Z\|_F$ and $\beta = c_2 \|X\|_F / \|S\|_F$ respectively, where c_i is a parameter which adjusts the contribution for each directed graph.

Experiments

In this section, we compare CoDiNMF with Spectral co-clustering (Sco) (Dhillon 2001), LP-FNMTF (Wang et al. 2011), and Co-clustering without directed graphs using SymNMF (CoNMF) which is a degenerate version of CoDiNMF.

Clustering Quality on US Patent and BlogCatalog Data Sets

We first apply our method to the US patent data set obtained from PatentsView[1] and BlogCatalog data set from (Wang et al. 2010). For US patent data set, we use the cooperative patent classification (CPC) info (as illustrated in Fig 3) to generate the ground truth clusters. The subset of US patent data sets which we used is displayed Table 1, and they have the citation information among patents, the connections between patents and words, but no word-word relationship information. The BlogCatalog data set also provides ground truth information, and it contains entity-entity, entity-feature, and feature-feature relations.

Table 2: Average F1 scores of co-clustering results

	A22	B06	B68	C14	BlogC
SCo	0.124	0.122	0.218	0.250	0.171
LP-FNMTF	0.234	0.165	0.285	0.285	0.144
CoNMF	0.242	0.205	0.307	0.303	0.249
CoDiNMF	**0.367**	**0.250**	**0.374**	**0.364**	**0.262**

Table 3: Rand index of co-clustering results

	A22	B06	B68	C14	BlogC
SCo	0.910	0.895	0.889	0.880	0.760
LP-FNMTF	0.977	0.935	0.938	0.901	0.759
CoNMF	0.976	0.931	0.941	0.886	0.766
CoDiNMF	**0.979**	**0.936**	**0.949**	**0.908**	**0.768**

Table 4: Adjusted Rand index of co-clustering results

	A22	B06	B68	C14	BlogC
SCo	0.028	0.098	0.124	0.110	0.063
LP-FNMTF	0.126	0.082	0.091	0.081	0.021
CoNMF	0.175	0.126	0.239	0.133	0.079
CoDiNMF	**0.246**	**0.169**	**0.275**	**0.151**	**0.082**

Table 5: Average number of co-clusters containing patents

patent class	A22	B06	B68	C14
ground truth	230	82	93	69
LP-FNMTF	230	82	93	69
SCo	124.0	51.2	56.2	49.2
CoNMF	208.0	74.0	92.0	65.5
CoDiNMF	222.2	77.3	92.0	68.0

To compare co-clusters with ground truth clusters, we treat each co-cluster as a cluster of patents by ignoring the terms in the co-cluster. We use average F1 score, Rand index and adjusted Rand index (ARI) which are well-known measures for accuracy. Specifically, the average F1 score for comparing clusters $\{V_1, \ldots, V_k\}$ and $\{G_1, \ldots, G_{k'}\}$ can be computed as

$$F_1 = \frac{1}{2k} \sum_{i=1}^{k} \max_j F_1(V_i, G_j) + \frac{1}{2k'} \sum_{j=1}^{k'} \max_i F_1(G_j, V_i),$$

where one of two cluster sets is the set of ground truth clusters (Du et al. 2017b). For CoDiNMF, we set the c_i parameters for balance parameter α and β with $c_i =$

[1] www.patentsview.org

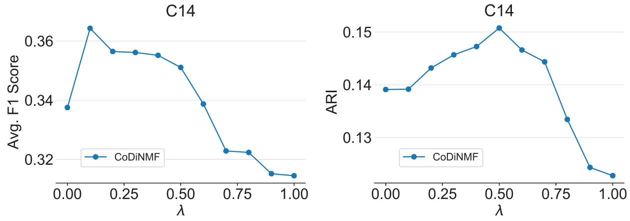

Figure 4: F1 score and ARI of CoDiNMF on C14 patent data set. There are 11 points corresponding to $\lambda \in \{0, 0.1, ..., 1\}$.

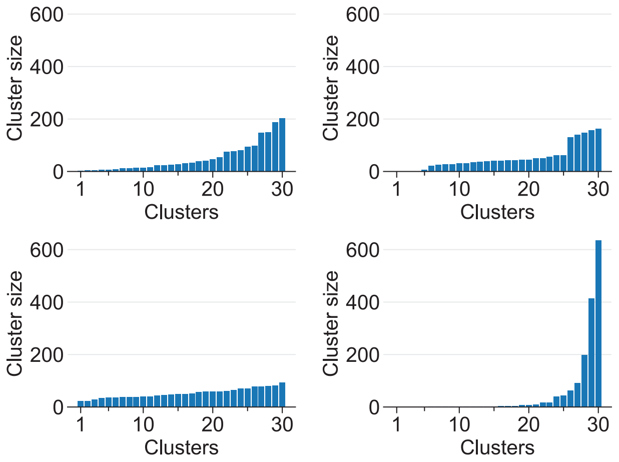

Figure 5: The number of philosophers in 30 co-clusters computed by CoDiNMF (the left upper figure), CoNMF (the right upper figure), LP-FNMTF(the left bottom figure) and SCo (the right bottom figure).

$\{1, 1.3, 1.7, 2, 3\}$. For CoNMF, we set S and Z as zero matrices, since it is a degenerate version of CoDiNMF.

The accuracies of four methods are compared in terms of F1 scores, Rand index, and adjusted Rand index in Table 2, Table 3, and Table 4 respectively. For the experimental results of LP-FNMTF on the patent data sets, we display the best scores between the results of LP-FNMTF using graph generated by k-nearest neighbor and using the given citation graph. The experimental results show that CoDiNMF consistently outperform other co-clustering methods in terms of three measures on the five data sets in Table 2, Table 3, and Table 4. Furthermore, F1 score and ARI of CoDiNMF are much better than CoNMF in Table 2 and Table 4. Thus, we can conclude that CoDiNMF improves the clustering accuracy by using the additional directed subgraphs by considering Table 2, Table 3, and Table 4.

Table 5 displays the average numbers of co-clusters containing patents for four methods. Comparing the ground truth and the results, we notice that many co-clusters computed by spectral co-clustering contain only the words. That is, the number of computed patent clusters is much less than the specified k. Although LP-FNMTF computes the same number of co-clusters with the ground truth, the clustering accuracies of CoDiNMF are much higher than LP-FNMTF, and the number of co-clusters computed by CoDiNMF is almost similar with the ground truth.

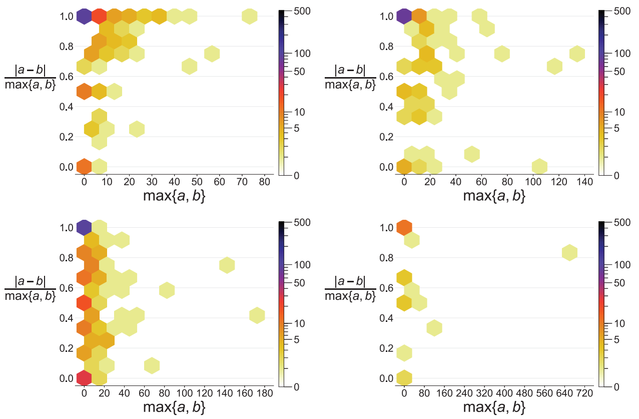

Figure 6: Distribution of maximum flow $\max\{a, b\}$ against relative ratio of net flow $\frac{|a-b|}{\max\{a,b\}}$ between clusters of philosophers computed by CoDiNMF (the left upper figure), CoNMF (the right upper figure), LP-FNMTF (the left bottom figure) and SCo (the right bottom figure), where a and b are opposite directional flow between two co-clusters. The color of each point represents the number of cluster pairs that fall into the point. The higher proportion of cluster pairs in the upper area in each figure, the more net directional flow among computed co-clusters is.

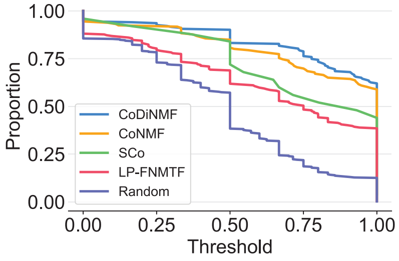

Figure 7: The proportion of cluster pairs of philosophers which are in the upper area above the threshold in Fig 6 is computed and displayed.

Finally, Fig 4 shows that F1 score and ARI of CoDiNMF can be improved by using an additional term which controls the sum of net directional flows among k co-clusters. Specifically, $\lambda \in \{0, 0.1, ..., 1\}$ in Eqn (13) adjusts the effects of additional term, and F1 scores and ARI of CoDiNMF in Fig 4 peak when λ is nonzero rather than $\lambda = 0$.

Case Study on Wikipedia Philosophers Data Set

We constructed the philosophers data set from a database snapshot of English Wikipedia[2] dumped on April 20th 2017.

[2] See the alphabetical index under https://en.wikipedia.org/wiki/Lists_of_philosophers

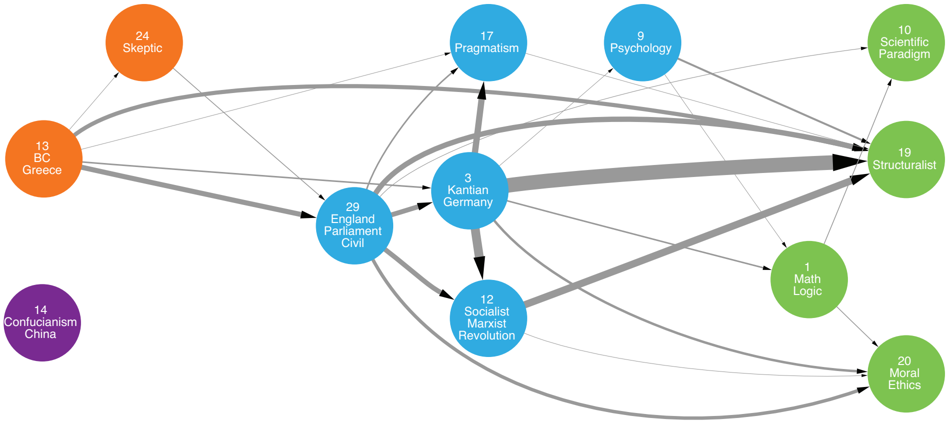

Figure 8: Influence network on the Wikipedia data set of philosophers. It is constructed by using net directional flows and topics computed by CoDiNMF. Each circle represents a co-cluster, and the number in the circle is the co-cluster id. The words in the circles are top keywords of the co-cluster, which are computed by CoDiNMF. The width of the arrows between co-clusters is proportional to the amount of net influence flow computed by CoDiNMF. We use top keywords of co-clusters to guess the era of philosophy of co-clusters, and we assign a color to each group of co-clusters by considering the era of philosophy.

Specifically, we define directed edges by using the influence relations among philosophers, which can be obtained in the infobox of most philosopher wiki articles and it is described in the supplementary information of (Ahn, Bagrow, and Lehmann 2010). We also use the contents of Wikipedia articles of each philosophers as the source of text information. In our data set, there are 1,479 philosophers and 13,002 directed edges among them. We run SCo, LP-FNMTF, CoNMF, and CoDiNMF to compute 30 co-clusters for this data set, and set $\lambda = 0.3$ for CoDiNMF.

First, we estimate the sizes of the clusters of philosophers given the computed co-clusters, and the corresponding results are displayed in Fig 5. We can observe that unlike CoDiNMF and LP-FNMTF, SCo generates about 20 empty or almost-empty clusters of philosophers. That is, SCo computes many co-clusters which contain only terms, and such co-clusters are not meaningful.

Next, we measure the quality of net flow by considering the relative ratio of net flow which can be defined as $|a - b|/\max\{a, b\}$, where a and b are opposite directional flow between two co-clusters. Intuitively, the larger the value of $|a - b|/\max\{a, b\}$ is, the clearer the directional flow between the two clusters is. Suppose that $a = 20$ and $b = 2$, then the relative ratio is $0.9 = (|20 - 2|)/20$, and we can think there is clear net directional flow. But, for $a = 18$ and $b = 20$, then the relative ratio is $0.1 = (|18 - 20|)/20$, and the net flow is not clearly directed. Fig 6 displays the distribution of the relative ratio of net flow between co-clusters, and it shows that SCo hardly generates enough number of clusters of philosophers, and also there are only few directional relations among the co-clusters despite of $k = 30$. Meanwhile, the distribution of LP-FNMTF displayed in Fig 6 is a little bit similar with the distribution of random. We can also notice that many cluster pairs in the results of CoDiNMF have high relative ratio of net flow compared to other co-clustering methods. Furthermore, we can observe that for CoDiNMF, no pair of clusters falls into the right lower corner of the plot while LP-FNMTF have several pairs of clusters very close to the right lower corner, which stands for poor directional flow. These observations demonstrate the effect of controlling term (Eqn (7)), which means that CoDiNMF is able to computes the k co-clusters which have many clear net directional flow among them. To support this assertion with the quantitative analysis, we provide Fig 7, and it displays the proportion of cluster pairs of philosophers which are in the upper area above the threshold of $|a - b|/\max\{a, b\}$ in Fig 6. Since CoDiNMF has always high proportion regardless of threshold, we can conclude that CoDiNMF is superior to find clear net directional flow among co-clusters compared to other co-clustering methods.

Table 6: Selected topics and philosophers of several co-clusters among top 10 keywords

1	Frege	Quine	mathematics	logic
3	Kant	Fichte	Kantian	Germany
12	socialist	Marx	Engels	revolutionary
19	Sartre	Lacan	structuralist	psychoanalysis
29	Locke	Hobbes	England	parliament

Finally, we provide a influence network in Fig 8, which is constructed by using the net directional flow between co-clusters and the topics of each co-cluster computed by CoDiNMF. The top topics and philosophers of co-clusters are selected by finding entries which have high values in the co-cluster indicator vector u_i, and some of them are summarized in Fig 8 and Table 6. To draw a graph, we ignore very small net flow for visual aid, and also exclude co-clusters with unusual sizes in the figure, since co-clusters with very small sizes are probably outliers and co-clusters with very large sizes are sometimes coarse. We notice that Fig 8 is consistent with real development of philosophy. For example, it shows that the philosophies in Greece affect various other philosophies; the philosophies in England and Germany have important roles in the era of modern philosophy, and they heavily affect contemporary philosophies; Especially, the structuralist and socialist are heavily influenced by Kantian. Another interesting point is that we can hardly find the meaningful net directional flow between the co-cluster of Chinese philosophy and other co-clusters of western philosophy. This is expected since Chinese philosophy were developed independently with western philosophy.

Conclusion

In this paper, we proposed a new co-clustering method, called CoDiNMF, which computes co-clusters of directed graphs. We designed the objective function of co-clustering by using min-cut problem and an additional term which controls net directional flow on directed subgraphs, and derived CoDiNMF which solves the proposed objective function. The experimental results showed that CoDiNMF improves the clustering quality compared to the other co-clustering method in terms of average F1 score, Rand index, and adjusted Rand index. Furthermore, we found directional flow of influences among co-clusters of philosophers by using CoDiNMF on the Wikipedia data set of philosophers. Our future work includes extensions of this framework to analyze hierarchical structures of health care data sets and Kiva data set which has many directed subgraphs among multiple entities.

Acknowledgments

W. Lim acknowledges support from 2016-2017 Google Ph.D. Fellowship in machine learning. This work was supported in part by the National Science Foundation (NSF) grant IIS-1348152 and Defense Advanced Research Projects Agency (DARPA) XDATA program grant FA8750-12-2-0309. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF or DARPA.

References

Ahn, Y.-Y.; Bagrow, J. P.; and Lehmann, S. 2010. Link communities reveal multiscale complexity in networks. *Nature* 466(7307):761–764.

Chen, X.; Ritter, A.; Gupta, A.; and Mitchell, T. 2015. Sense discovery via co-clustering on images and text. In *Proceedings of CVPR*, 5298–5306.

Cho, H.; Dhillon, I. S.; Guan, Y.; and Sra, S. 2004. Minimum sum-squared residue co-clustering of gene expression data. In *Proceedings of SDM*, 114–125. SIAM.

Dhillon, I. S.; Mallela, S.; and Modha, D. S. 2003. Information-theoretic co-clustering. In *Proceedings of the SIGKDD*, 89–98. ACM.

Dhillon, I. S. 2001. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the SIGKDD*, 269–274. ACM.

Du, R.; Kuang, D.; Drake, B.; and Park, H. 2017a. DC-NMF: nonnegative matrix factorization based on divide-and-conquer for fast clustering and topic modeling. *Journal of Global Optimization* 1–22.

Du, R.; Kuang, D.; Drake, B.; and Park, H. 2017b. Hierarchical community detection via rank-2 symmetric nonnegative matrix factorization. *Computational Social Networks*.

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

Kuang, D., and Park, H. 2013. Fast rank-2 nonnegative matrix factorization for hierarchical document clustering. In *SIGKDD*, 739–747. ACM.

Kuang, D.; Yun, S.; and Park, H. 2015. SymNMF: Nonnegative low-rank approximation of a similarity matrix for graph clustering. *Journal of Global Optimization* 62(3):545–574.

Long, B.; Zhang, Z. M.; and Yu, P. S. 2005. Co-clustering by block value decomposition. In *Proceedings of SIGKDD*, 635–640. ACM.

Nie, F.; Li, J.; and Li, X. 2017. Self-weighted multiview clustering with multiple graphs. In *Proceedings of IJCAI*.

Rohe, K.; Qin, T.; and Yu, B. 2016. Co-clustering directed graphs to discover asymmetries and directional communities. *PNAS* 113(45):12679–12684.

Sun, J.; Lu, J.; Xu, T.; and Bi, J. 2015. Multi-view sparse co-clustering via proximal alternating linearized minimization. In *Proceedings of ICML*, 757–766.

Tang, W.; Lu, Z.; and Dhillon, I. S. 2009. Clustering with multiple graphs. In *Proceedings of ICDM*, 1016–1021. IEEE.

Vlachos, M.; Fusco, F.; Mavroforakis, C.; Kyrillidis, A.; and Vassiliadis, V. G. 2014. Improving co-cluster quality with application to product recommendations. In *Proceedings of CIKM*, 679–688. ACM.

Wang, X.; Tang, L.; Gao, H.; and Liu, H. 2010. Discovering overlapping groups in social media. In *Proceedings of ICDM*, 569–578. IEEE.

Wang, H.; Nie, F.; Huang, H.; and Makedon, F. 2011. Fast nonnegative matrix tri-factorization for large-scale data co-clustering. In *Proceedings of IJCAI*, volume 22, 1553.

Wu, T.; Benson, A. R.; and Gleich, D. F. 2016. General tensor spectral co-clustering for higher-order data. In *Proceedings of NIPS*, 2559–2567.

Xu, W.; Liu, X.; and Gong, Y. 2003. Document clustering based on non-negative matrix factorization. In *Proceedings of SIGIR*, 267–273. ACM.

Zhu, Y.; Yang, H.; and He, J. 2015. Co-clustering based dual prediction for cargo pricing optimization. In *Proceedings of SIGKDD*, 1583–1592. ACM.