Subspace Co-clustering with Two-Way Graph Convolution

Chakib Fettal
[1]Centre Borelli, Université Paris Cité
[2]Informatique CDC
chakib.fettal@etu.u-paris.fr

Lazhar Labiod
Centre Borelli UMR 9010
Université Paris Cité
lazhar.labiod@u-paris.fr

Mohamed Nadif
Centre Borelli UMR 9010
Université Paris Cité
mohamed.nadif@u-paris.fr

ABSTRACT

Subspace clustering aims to cluster high dimensional data lying in a union of low-dimensional subspaces. It has shown good results on the task of image clustering but text clustering, using document-term matrices, proved more impervious to advances based on this approach. We hypothesize that this is because, compared to image data, text data is generally higher dimensional and sparser. This renders subspace clustering impractical in such a context. Here, we leverage subspace clustering for text by addressing these issues. We first extend the concept of subspace clustering to co-clustering, which has been extensively used on document-term matrices due to the resulting interplay between the document and term representations. We then address the sparsity problem through a two-way graph convolution, which promotes the grouping effect that has been credited for the effectiveness of some subspace clustering models. The proposed formulation results in an algorithm that is efficient both in terms of computational and spatial complexity. We show the competitiveness of our model *w.r.t* the state-of-the-art on document-term attributed graph datasets in terms of performance and efficiency.

CCS CONCEPTS

• **Information systems → Clustering**.

KEYWORDS

Co-clustering, Subspace Clustering, Subspace Co-clustering, Attributed Graphs, Text mining

ACM Reference Format:
Chakib Fettal, Lazhar Labiod, and Mohamed Nadif. 2022. Subspace Co-clustering with Two-Way Graph Convolution. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM '22), October 17–21, 2022, Atlanta, GA, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3511808.3557706

1 INTRODUCTION AND BACKGROUND

Subspace clustering [18] is an unsupervised problem where one wishes to group points according to the subspaces in which they lie. There have been a variety of approaches to solve the problem, a lot of which consider the self-expressive formulation where it is assumed that each element can be written as a linear combination of the elements in the that subspace. Typically, the generic formulation is given as

$$\min_{\mathbf{R}} \quad \| \mathbf{X} - \mathbf{R}\mathbf{X} \|^2 + \Omega(\mathbf{R}) \quad \text{s.t.} \quad \mathbf{R} \in \mathcal{R} \tag{1}$$

where $\mathbf{X} \in \mathbb{R}^{n \times d}$ is a matrix of d-dimensional data points, $\mathbf{R} \in \mathbb{R}^{n \times n}$ is called the self-representation matrix, $\Omega(\mathbf{R})$ serves as a regularization term to induce desirable properties on $\mathbf{R}$ and avoid trivial solutions (such as $\mathbf{R} = \mathbf{I}$), and $\mathcal{R}$ is the feasible region.

After finding an optimal solution $\mathbf{R}^*$, an affinity matrix is generated based on the magnitudes of the entries of $\mathbf{R}^*$, usually using $|\mathbf{R}^* + \mathbf{R}^{*\top}|/2$, and a partition of the points is then generated using a graph clustering method e.g. the spectral clustering algorithm [23].

Subspace clustering methods based on the self-expressive property [29] have been widely used to cluster image datasets due the assumption that image datasets are often drawn from multiple low-dimensional subspaces. One of the earlier approaches was the least-square regression (LSR) subspace clustering [14] that leverages a grouping effect based on the correlation of the data to do the segmentation. More sophisticated approaches that make up the state-of-the-art for proposed were later proposed such as the Elastic-net Subspace Clustering (EnSC) [27] and the subspace clustering through the orthogonal matching pursuit (SSC-OMP) [28]. However, despite text data also fulfilling this assumption, to the best of our knowledge, no self-expressive subspace clustering approach specifically tailored to text has been proposed. This can perhaps be explained by the fact that document-term datasets are usually much larger and sparser than image datasets and thus each individual data point could potentially lie in a unique subspace.

In this paper we propose a subspace clustering model tailored for document-term matrices through the concept of co-clustering i.e. using the interplay between rows and columns, or in the context of text, the interplay between documents and terms to generate a segmentation for both of them. We also propose a way to overcome the possible problem of each document/term lying in a unique subspace through using a two-way graph convolution that consists of a weighted Laplacian smoothing preprocessing step inspired by the simple graph convolutional network [5, 12, 25].

2 PROPOSED METHOD

Notation. Matrices are denoted with boldface uppercase and vectors with boldface lowercase letters. Given a matrix $\mathbf{X}$, its i-th row is denoted by $\mathbf{x}_i$ and its j-th column by $\mathbf{x}'_j$. $\mathbf{I}_n$ is the identity matrix of size n. The Frobenius norm is denoted by $\|.\|$. k and g denote the number of row and column clusters. Function $[\mathbf{U}, \Sigma, \mathbf{V}] = \text{SVD}(\mathbf{X})$ gives the singular value decomposition of matrix $\mathbf{X}$ where $\mathbf{U}$ and $\mathbf{V}$ are the left and right singular vectors and Σ is the diagonal matrix containing the singular values, sorted in decreasing order.

CIKM '22, October 17–21, 2022, Atlanta, GA, USA

ACM ISBN 978-1-4503-9236-5/22/10…$15.00
https://doi.org/10.1145/3511808.3557706

2.1 Self-Expressive Subspace Co-clustering

The advantage of co-clustering [8] is that it makes use of the inherent duality between the rows and columns of data tables which can lead to improvement in partitioning along both dimensions. For example, in the case of the document term matrices [1, 2, 9, 17, 19], co-clustering incorporates term space information to do the document partitioning and vice-versa. Motivated by this observation, given a document-term matrix $\mathbf{X} \in \mathbb{R}_+^{n \times d}$, we formulate the subspace co-clustering problem as

$$\min_{\mathbf{R},\mathbf{C}} \quad \| \mathbf{X} - \mathbf{RXC} \|^2 + \Omega(\mathbf{R}, \mathbf{C}) \quad \text{s.t.} \quad \mathbf{R} \in \mathcal{R},\ \mathbf{C} \in C \tag{2}$$

where $\mathbf{R} \in \mathbb{R}^{n \times n}$ and $\mathbf{C} \in \mathbb{R}^{d \times d}$ are respectively the row and column self-representation matrices, $\Omega(\mathbf{R}, \mathbf{C})$ is the regularization term where the regularization of $\mathbf{R}$ and $\mathbf{C}$ can either be independent i.e. $\Omega(\mathbf{R}, \mathbf{C}) = \Omega_{\mathbf{R}}(\mathbf{R}) + \Omega_{\mathbf{C}}(\mathbf{C})$ or dependant, $\mathcal{R}$ and C are the feasible regions.

2.2 Promoting the Grouping Effect Through a Two-way Graph Convolution

Some subspace clustering methods [11, 13, 14] ascribe the performance of their clustering to the grouping effect.

Definition (Grouping Effect): Given a data matrix $\mathbf{X}$, a self-representation matrix $\mathbf{R}$ has a grouping effect if

$$\forall i \neq j,\ \|\mathbf{x}_i - \mathbf{x}_j\|^2 \to 0 \implies \|\mathbf{r}_i - \mathbf{r}_j\|^2 \to 0.$$

This could pose a problem in the case of text because of its high dimensionality and sparsity as data points may not be sufficiently "close" in the sense of the self-expressive property to be grouped in a meaningful way. This means data points need some sort of smoothing to help subspace clustering algorithms to find common subspaces. We propose to solve this problem through a two-way graph convolution. This requires two similarity matrices that will act as graphs on the rows $\mathbf{S}_{\mathbf{R}}$ and columns $\mathbf{S}_{\mathbf{C}}$. These matrices can either be constructed through some similarity measure on the data or be provided a priori, e.g. in the case of attributed graphs.

An intuition can be drawn from the fact that the rows and columns of $\mathbf{S}_{\mathbf{R}}^p \mathbf{X} \mathbf{S}_{\mathbf{C}}^q$, as propagation orders p, q grow, are getting smoother by being averaged up to their p-th and q-th neighbors respectively akin to Laplacian smoothing. This operation thus makes the rows and columns more and more similar. Trough the grouping property this implies that the self-representation vectors should also be getting more similar, leading to a more meaningful partitioning.

The challenge is to choose suitable propagation orders since large values can cause over-smoothing and make all data points look similar. Our problem becomes

$$\min_{\mathbf{R},\mathbf{C}} \quad \left\| \mathbf{S}_{\mathbf{R}}^p \mathbf{X} \mathbf{S}_{\mathbf{C}}^q - \mathbf{R} \left(\mathbf{S}_{\mathbf{R}}^p \mathbf{X} \mathbf{S}_{\mathbf{C}}^q \right) \mathbf{C} \right\|^2 + \Omega(\mathbf{R}, \mathbf{C}) \text{ s.t. } \mathbf{R} \in \mathcal{R},\ \mathbf{C} \in C. \tag{3}$$

In what follows, we will refer to the smoothed matrix $\mathbf{S}_{\mathbf{R}}^p \mathbf{X} \mathbf{S}_{\mathbf{C}}^q$ using $\mathbf{H}$ since this operation can be considered as a sort of data preprocessing step, independent from the clustering model. Note that the complexity of this operation is in $O(p\|\mathbf{S}_{\mathbf{R}}\|_0 + q\|\mathbf{S}_{\mathbf{C}}\|_0)$ where $\|.\|_0$ is the 0-norm that gives the number of non-zero entries of its input.

2.3 Subspace Co-clustering through LSR

We propose an initial variant based on the LSR subspace clustering model where we define the regularization term as follows $\Omega(\mathbf{R}, \mathbf{C}) = \lambda_{\mathbf{R}}\|\mathbf{R}\|^2 + \lambda_{\mathbf{C}}\|\mathbf{C}\|^2$ where $\lambda_{\mathbf{R}}$ and $\lambda_{\mathbf{C}}$ are parameters that regulate the trade-off between the reconstruction term and the regularizer. The formulation of the problem becomes

$$\min_{\mathbf{R},\mathbf{C}} \quad \|\mathbf{H} - \mathbf{RHC}\|^2 + \lambda_{\mathbf{R}}\|\mathbf{R}\|^2 + \lambda_{\mathbf{C}}\|\mathbf{C}\|^2. \tag{4}$$

By fixing $\mathbf{R}$ and solving for $\mathbf{C}$ and inversely, a closed form solution can be obtained for both matrices where we can explicitly see how our model uses information from the columns for the row space partitioning and vice-versa

$$\begin{aligned} \mathbf{R} &= \mathbf{H}\mathbf{C}^\top\mathbf{H}^\top \left(\mathbf{H}\mathbf{C}\mathbf{C}^\top\mathbf{H}^\top + \lambda_{\mathbf{R}}\mathbf{I}\right)^{-1} \\ \mathbf{C} &= \left(\mathbf{H}^\top\mathbf{R}^\top\mathbf{R}\mathbf{H} + \lambda_{\mathbf{C}}\mathbf{I}\right)^{-1} \mathbf{H}^\top\mathbf{R}^\top\mathbf{H}. \end{aligned} \tag{5}$$

However, solving the problem requires an iterative process where we alternatively fix one of $\mathbf{R}$ and $\mathbf{C}$ and update the other until convergence. The overall computational complexity is $O(n^3 + d^3 + tnd^2 + tn^2d)$, where t is the number of iterations, and the spatial complexity is $O(n^2 + d^2)$, which is prohibitive for a lot of real world applications.

2.4 A More Efficient Formulation Through Orthogonality Constraints

To address the issue of complexity we propose to introduce the following constraints $\mathbf{R} = \mathbf{Z}\mathbf{Z}^\top$ and $\mathbf{C} = \mathbf{W}\mathbf{W}^\top$ where $\mathbf{Z} \in \mathbb{R}^{n \times k}$ and $\mathbf{W} \in \mathbb{R}^{d \times g}$ are semi-orthogonal i.e. $\mathbf{Z}^\top\mathbf{Z} = \mathbf{I}_k$ and $\mathbf{W}^\top\mathbf{W} = \mathbf{I}_g$. With these constraints the problem becomes simpler due to fact that $\|\mathbf{Z}\|^2 = \mathtt{rank}(\mathbf{Z})$ and $\|\mathbf{W}\|^2 = \mathtt{rank}(\mathbf{W})$. The new formulation of the problem is

$$\min_{\mathbf{Z},\mathbf{W}} \quad \|\mathbf{H} - \mathbf{Z}\mathbf{Z}^\top\mathbf{H}\mathbf{W}\mathbf{W}^\top\|^2 \quad \text{s.t.} \quad \mathbf{Z}^\top\mathbf{Z} = \mathbf{I}_k,\ \mathbf{W}^\top\mathbf{W} = \mathbf{I}_g \tag{6}$$

At first glance this problem also requires an alternating solving scheme using two update rules that we obtain by fixing $\mathbf{W}$ and solving for $\mathbf{Z}$ and vice versa

$$\begin{aligned} \mathbf{Z} &= [\mathbf{u}'_1, \ldots, \mathbf{u}'_k] \quad \text{s.t.} \quad [\mathbf{U}, \Sigma, \mathbf{V}] = \text{SVD}(\mathbf{HW}) \\ \mathbf{W} &= [\mathbf{u}'_1, \ldots, \mathbf{u}'_k] \quad \text{s.t.} \quad [\mathbf{U}, \Sigma, \mathbf{V}] = \text{SVD}(\mathbf{H}^\top\mathbf{Z}). \end{aligned} \tag{7}$$

This entails that $g = k$. The detailed pseudo-code for this method is given in algorithm 1, its spatial complexity is the same as for LSR but the computational complexity is in $O(n^3 + d^3 + tnd\log(k))$, which is faster. All in all, the approach still remains quite inefficient.

Efficient computation of $\mathbf{Z}^*$ and $\mathbf{W}^*$. The previous problem can be efficiently solved using a single truncated SVD. This is a consequence of the following proposition.

Proposition 1. The alternating process defined in system of (7) converges to $\mathbf{Z}$ and $\mathbf{W}$ being the left and right truncated singular vectors of $\mathbf{H}$ respectively.

This results in a more efficient algorithm since we circumvent the iterative step. However, the interaction between rows and columns implicitly remains since the resulting solution is also a solution to the aforementioned alternating optimization problem where the interaction is explicit.

Algorithm 1: Naive SCC

Input : $\mathbf{X}$ data matrix, $\mathbf{S_R}$ and $\mathbf{S_C}$ row and column prop. matrices, p and q row and column prop. orders, k number of co-clusters

Output: $\mathbf{R}$ and $\mathbf{C}$ the row and column self-representation matrices, $\pi_\mathbf{R}$ and $\pi_\mathbf{C}$ row and columns partitions

$\mathbf{H} \leftarrow \mathbf{S_R}^p \mathbf{X} \mathbf{S_C}^q$;

$\mathbf{W} \leftarrow \mathbf{W}_{init}$;

while *not converged* **do**

  $\mathbf{Z} \leftarrow [\mathbf{u}'_1, \ldots, \mathbf{u}'_k]$ s.t. $[\mathbf{U}, \_, \_] \leftarrow \text{SVD}(\mathbf{HW})$;

  $\mathbf{W} \leftarrow [\mathbf{u}'_1, \ldots, \mathbf{u}'_k]$ s.t. $[\mathbf{U}, \_, \_] \leftarrow \text{SVD}(\mathbf{H}^\top \mathbf{Z})$;

end

$\mathbf{R}, \mathbf{C} \leftarrow \mathbf{Z}\mathbf{Z}^\top, \mathbf{W}\mathbf{W}^\top$;

Generate $\pi_\mathbf{R}$, $\pi_\mathbf{C}$ through spectral clustering on $|\mathbf{R}|$ and $|\mathbf{C}|$;

Proof. Suppose, without loss of generality, that $k \leq \text{rank}(\mathbf{H})$. We have that $k = \text{rank}(\mathbf{Z}) = \text{rank}(\mathbf{W})$ implying

$$\text{rank}\left(\mathbf{Z}\mathbf{Z}^\top \mathbf{H}\mathbf{W}\mathbf{W}^\top\right) \leq k.$$

This means that we are looking for the best k-rank approximation. Given $[\mathbf{U}, \Sigma, \mathbf{V}] = \text{SVD}(\mathbf{H})$, by setting $\mathbf{Z} = \mathbf{U}_k = [\mathbf{u}'_1, \ldots, \mathbf{u}'_k]$ and $\mathbf{W} = \mathbf{V}_k = [\mathbf{v}'_1, \ldots, \mathbf{v}'_k]$. We have that

$$\begin{aligned} \|\mathbf{H} - \mathbf{Z}\mathbf{Z}^\top \mathbf{H}\mathbf{W}\mathbf{W}^\top\|^2 &= \|\mathbf{H} - \mathbf{U}_k \mathbf{U}_k^\top \mathbf{U}\Sigma\mathbf{V}^\top \mathbf{V}_k \mathbf{V}_k^\top\|^2 \\ &= \|\mathbf{H} - [\mathbf{U}_k, \mathbf{0}]\Sigma[\mathbf{V}_k, \mathbf{0}]^\top\|^2 \\ &= \|\mathbf{H} - \mathbf{U}_k \Sigma_k \mathbf{V}_k^\top\|^2 \end{aligned} \tag{8}$$

which according to the Eckart–Young–Mirsky theorem is the optimal value of the rank-k approximation problem of $\mathbf{H}$. □

From the previous result, we can show that our approach has an approximate grouping effect

Proposition 2. Given matrix $\mathbf{H}$, the solutions $\mathbf{R}$ and $\mathbf{C}$ display a grouping effect on matrix $\tilde{\mathbf{H}}$, the best rank-k approximation of $\mathbf{H}$, since $\mathbf{z}_i = \Sigma_k^{-1} \mathbf{W}^\top \tilde{\mathbf{h}}_i$ and $\mathbf{w}_i = \Sigma_k^{-1} \mathbf{Z}^\top \tilde{\mathbf{h}}_i'^\top$.

Efficient Spectral Clustering from $\mathbf{R}^*$ ***and*** $\mathbf{C}^*$**.** The optimal coefficient matrix $\mathbf{R}^* = \mathbf{Z}^*\mathbf{Z}^{*\top}$ is symmetric by construction, however its entries are not necessarily nonnegative which implies having to use element-wise absolute value to obtain a valid affinity matrix. This would destroy all information we already have on the decomposition of $\mathbf{R}^*$ into $\mathbf{Z}^*\mathbf{Z}^{*\top}$ since generally there is no relation between the spectrum of a matrix and its spectrum after applying an entrywise function. We circumvent this problem by instead considering affinity matrix $\mathbf{K_R} = (r_{ij} + 1)^2_{ij}$. We thus have that $\mathbf{K_R} = \langle \varphi(\mathbf{Z}^*), \varphi(\mathbf{Z}^*) \rangle$ where φ is a the feature map for the second degree polynomial kernel $\mathbf{K_R}$ applied on the row vectors of $\mathbf{Z}$ i.e.

$$\begin{aligned} \varphi(\mathbf{z}) = \langle & z_k^2, \ldots, z_1^2, \sqrt{2} z_k z_{k-1}, \ldots, \sqrt{2} z_k z_1, \sqrt{2} z_{k-1} z_{k-2}, \\ & \ldots, \sqrt{2} z_{k-1} z_1, \ldots, \sqrt{2} z_2 z_1, \sqrt{2} z_k, \ldots, \sqrt{2} z_1, 1 \rangle \end{aligned} \tag{9}$$

with $\varphi : \mathbb{R}^k \to \mathbb{R}^{\binom{k+2}{2}}$. Any feature map for an entrywise nonnegative kernel is a possible alternative. We chose the simplest exact feature map possible here since the transformation does not result in too much of a dimensionality increase for the inputs as $k << n, d$. Approximations can be used in order to work with otherwise infinite dimensional feature map kernels e.g. the RBF kernel. We then propose to perform the spectral clustering directly on the affinity matrix instead of the Graph Laplacian as in [21]. Since the eigenvectors of $\mathbf{K_R}$ are the same as the left singular vectors of $\varphi(\mathbf{Z}^*)$ the process is much faster since $k << d$. To obtain a clustering from $\mathbf{C}^*$, the operations are the same. The overall computation complexity is then in $O\left((nd + nk^2 + dk^2)\log(k)\right)$ while the spatial one is in $O\left(nk^2 + dk^2\right)$.

3 EXPERIMENTS

3.1 Experimental Setup

Datasets. We use four attributed graph citation networks, which are graphs characterized by an adjacency matrix $\mathbf{A}$ and a node features matrix $\mathbf{X}$. The summary statistics are available in table 2.

Table 2: Dataset statistics.

Dataset	#Nodes	#Edges	#Features	#Classes
ACM [24]	3025	9150593	1870	3
CiteSeer [22]	3327	4732	3703	6
PubMed [22]	19717	44338	500	3
Wiki [26]	2405	17981	4973	17

Comparative models. We compare our model to clustering and co-clustering models that either use only the input node feature matrix $\mathbf{X}$ or that use both $\mathbf{A}$ and $\mathbf{X}$ i.e. attributed graph clustering/co-clustering models.

- *Vanilla clustering models.* The k-means algorithm is our baseline.
- *Subspace clustering models.* We also use the aforementioned subspace clustering LSR, EnSC and SSC-OMP models.
- *Attributed graph clustering models.* We use GIC [16] where clustering is done by maximizing the mutual information between nodes contained in the same cluster, AGE [4] proposes a Laplacian smoothing filter that acts as a low-pass filter applied in adaptive learning scheme, S^{2}GC [30] proposes a method for the aggregation of K-hop neighborhoods that is a trade-off of low- and high-pass filter bands, and GCC [7] which proposes a simultaneous representation learning and clustering scheme for nodes.
- *Vanilla co-clustering models.* We compare the model to the spectral co-clustering algorithm [6] and the DCC [20] which is based on a regularized von Mises-Fisher mixture model.
- *Attributed graph co-clustering Models.* The only such model is CFOND [10], a consensus factorization model that simultaneously factorizes information from three aspects: network topology structures, instance-feature content relationships, and feature-feature correlations.

Experimental settings. For our method, we use as our row graph, the adjacency matrix provided in the datasets $\mathbf{S_R} = \mathbf{A}$. For the columns, we use $\mathbf{S_C} = \left(\max\left\{\log\left(\frac{\mathbf{C}_{\cdot\cdot}}{\mathbf{C}_{i\cdot}\mathbf{C}_{\cdot j}} c_{ij}\right), 0\right\}\right)_{ij}$ where $\mathbf{C} = \mathbf{X}^\top\mathbf{X}$, which is the nonnegative pointwise mutual information [3] matrix of the terms; Intuitively $s_{\mathbf{C}ij}$ gives the semantic relatedness of term i and j, the larger the value, the more these terms are related. Both matrices are then added self-loops and normalized

Table 1: Clustering performance on the datasets averaged over ten runs. The best results are highlighted in bold. Our model is competitive with the state-of-the-art as it has the best results on most datasets while having small standard deviations.

Method	Input	ACM			CiteSeer			PubMed			Wiki		
		Acc	NMI	ARI	Acc	NMI	ARI	Acc	NMI	ARI	Acc	NMI	ARI
k-means	**X**	62.8±4.8	37.2±9.2	34.5±10.4	62.5±1.6	36.7±1.9	35.5±2.5	60.1±0.0	31.4±0.0	28.1±0.0	47.3±6.0	46.3±6.9	26.4±8.1
LSR	**X**	80.3±0.0	47.0±0.0	51.9±0.0	21.1±0.0	0.2±0.1	0.0±0.0		OOM		21.1±3.3	9.0±5.9	2.6±2.0
EnSC	**X**	79.5±0.0	46.8±0.0	50.3±0.0	55.6±0.0	14.8±0.0	14.6±0.0	55.6±0.0	14.8±0.0	14.7±0.0	45.5±2.0	45.7±1.7	28.8±1.3
SSC-OMP	**X**	78.8±0.1	43.4±0.1	48.3±0.1	24.0±1.1	3.5±0.4	1.8±0.1	60.4±0.0	22.3±0.0	19.4±0.0	52.7±4.4	48.1±2.3	33.3±1.5
Spectral	**X**	80.6±0.1	48.4±0.1	52.3±0.1	30.3±1.7	10.0±1.3	5.5±1.6	61.2±0.0	24.7±0.0	21.8±0.0	37.8±1.2	38.2±0.3	20.8±0.4
DCC	**X**	40.5±3.3	7.8±5.1	2.1±2.2	35.1±3.8	11.5±2.4	8.9±2.9	54.3±3.6	16.5±2.9	13.4±4.1	48.3±3.6	47.5±2.6	30.6±3.0
GIC	**A, X**	34.3±0.4	0.1±0.1	0.0±0.0	68.8±0.8	43.8±1.0	44.6±1.0	64.3±0.4	26.0± 0.5	23.6±0.5	46.5±1.4	48.2±0.5	30.2±1.4
S^2GC	**A, X**	40.5±3.4	1.7±1.2	1.8±1.3	68.1±0.3	42.3±0.2	43.5±0.3	70.8±0.0	**32.5±0.0**	**33.2±0.0**	52.7±1.0	49.0±0.3	29.6±0.9
GCC	**A, X**	35.4±0.0	0.3±0.0	0.0±0.0	69.4±0.1	**45.0±0.2**	**45.4±0.1**	70.8±0.0	32.3±0.0	**33.2±0.0**	54.1±0.8	**55.0±0.2**	**33.3±0.5**
CFOND	**A, X**	71.8±0.6	37.2±0.5	38.2±0.7	63.0±1.1	36.6±1.3	36.2±1.2	60.1±0.0	31.4±0.0	28.1±0.0	47.8±3.0	49.5±2.1	30.3±2.5
SCC	**A, X**	**81.4±0.0**	**50.0±0.0**	**53.9±0.0**	**69.5±0.0**	43.5±0.0	43.7±0.0	**70.9±0.0**	31.7±0.0	**33.2±0.0**	**59.3±0.6**	53.9±0.9	32.7±1.4

Table 3: Training times in seconds of the different subspace clustering models averaged over ten runs. Ours is the fastest one on all datasets.

Method	ACM	CiteSeer	PubMed	Wiki
ENSC	1395.9	405.2	1416	1447.2
SSC-OMP	168.0	263.4	1447	237.5
LSR	21.5	157.7	OOM	21.1
SCC	**9.7**	**7.4**	**28.9**	**8.8**

as in [7]. The row propagation order p is selected using the selection rule proposed in [7], while for the column one we set $q = 1$. We perform ten runs for each model. We use the author provided implementation and parameters when available. For models that use a parameter p like ours, we run their proposed selection rule until convergence with no maximum p specified, for fairness. All experiments were performed on the same machine.

3.2 Document Clustering

We compare the methods on document clustering using the Clustering Accuracy (Acc), Normalized Mututal Information (NMI) and Adjusted Rand Score (ARI) metrics. Table 1 shows the performance of the different models. Methods that use both graph structure and features outperform methods that use the node features only except on ACM where the graph is not informative (most entries are set to one), we used this dataset to show the robustness of our model in the face of uninformative graph structure compared to state of the art attributed graph clustering models. We see that our approach is competitive and outperforms other models on all datasets in terms of accuracy. It also has near zero standard deviation on most metrics which is a sign of robustness. We report in table 3 the training times of our algorithm compared to other subspace clustering models. We can see that our subspace clustering approach is faster in the different datasets by significant margins even though it generates a clustering for both rows and columns.

3.3 Term Clustering

Co-clustering models additionally generate a clustering for the terms. Since the PubMed dataset is the only one for which we managed to find the actual terms, we performed the term clustering solely on it. Table 4 presents the most frequent terms for each topic found by our model. The PubMed dataset contains scientific papers concerning diabetes. We see that topic *a* contains terms that are related to a presentation of diabetes, e.g, *insulin*, *glucos*, *type*, etc. Topic *b* has terms that are related to the microscopic effects of diabetes such as *cell*, *islet*, *gene*, *betacel*, and so on. Finally, topic *c* seems to concern terms that are associated with medical experimentation and analysis of results such as *control*, *increas*, *signific*, etc. We note the coherence of these term clusters since they cluster the PubMed paper contents according to three topics. This term clustering can then be used to help characterize document clusters to facilitate interpretation (for more details, see [15]).

Table 4: The three topics found by SCC characterized by their top ten most frequent terms.

Topic *a*	Topic *b*	Topic *c*
patient	cell	rat
insulin	mice	control
glucos	islet	activ
type	iddm	level
group	gene	increas
subject	diseas	respons
lt	develop	signific
risk	nod	effect
associ	children	express
treatment	betacel	plasma

4 CONCLUSION

We proposed SCC, a new approach to leverage subspace clustering for text data through co-clustering and two-way graph convolution. It circumvents the computational and spatial complexity issues of subspace clustering through using factor matrices and nonnegative kernel feature maps. Experiments showed that our model is competitive with the state of the art for attributed graph node clustering in terms of performance and robustness.

REFERENCES

[1] Séverine Affeldt, Lazhar Labiod, and Mohamed Nadif. 2020. Ensemble block co-clustering: a unified framework for text data. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 5–14.

[2] Séverine Affeldt, Lazhar Labiod, and Mohamed Nadif. 2021. Regularized Dual-PPMI Co-clustering for Text Data. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2263–2267.

[3] Kenneth Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational linguistics* 16, 1 (1990), 22–29.

[4] Ganqu Cui, Jie Zhou, Cheng Yang, and Zhiyuan Liu. 2020. Adaptive graph encoder for attributed graph embedding. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 976–985.

[5] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems* 29 (2016).

[6] Inderjit S Dhillon. 2001. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*. 269–274.

[7] Chakib Fettal, Lazhar Labiod, and Mohamed Nadif. 2022. Efficient Graph Convolution for Joint Node Representation Learning and Clustering. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. 289–297.

[8] Gérard Govaert and Mohamed Nadif. 2013. *Co-clustering: models, algorithms and applications*. John Wiley & Sons.

[9] Gérard Govaert and Mohamed Nadif. 2018. Mutual information, phi-squared and model-based co-clustering for contingency tables. *Advances in data analysis and classification* 12, 3 (2018), 455–488.

[10] Ting Guo, Shirui Pan, Xingquan Zhu, and Chengqi Zhang. 2018. CFOND: consensus factorization for co-clustering networked data. *IEEE Transactions on Knowledge and Data Engineering* 31, 4 (2018), 706–719.

[11] Han Hu, Zhouchen Lin, Jianjiang Feng, and Jie Zhou. 2014. Smooth representation clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3834–3841.

[12] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[13] Canyi Lu, Jiashi Feng, Zhouchen Lin, and Shuicheng Yan. 2013. Correlation adaptive subspace segmentation by trace lasso. In *Proceedings of the IEEE international conference on computer vision*. 1345–1352.

[14] Can-Yi Lu, Hai Min, Zhong-Qiu Zhao, Lin Zhu, De-Shuang Huang, and Shuicheng Yan. 2012. Robust and efficient subspace segmentation via least squares regression. In *European conference on computer vision*. Springer, 347–360.

[15] F Marcotorchino. 1987. Block seriation problems: A unified approach. Reply to the problem of H. Garcia and JM Proth (Applied Stochastic Models and Data Analysis, 1,(1), 25–34 (1985)). *Applied Stochastic Models and Data Analysis* 3, 2 (1987), 73–91.

[16] Costas Mavromatis and George Karypis. 2021. Graph InfoClust: Maximizing Coarse-Grain Mutual Information in Graphs. In *PAKDD (1)*. 541–553.

[17] Mohamed Nadif and François Role. 2021. Unsupervised and self-supervised deep learning approaches for biomedical text mining. *Briefings in Bioinformatics* 22, 2 (2021), 1592–1603.

[18] Lance Parsons, Ehtesham Haque, and Huan Liu. 2004. Subspace clustering for high dimensional data: a review. *Acm sigkdd explorations newsletter* 6, 1 (2004), 90–105.

[19] Aghiles Salah, Melissa Ailem, and Mohamed Nadif. 2018. Word co-occurrence regularized non-negative matrix tri-factorization for text data co-clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 3992–3999.

[20] Aghiles Salah and Mohamed Nadif. 2017. Model-based von mises-fisher co-clustering with a conscience. In *Proceedings of the 2017 SIAM International Conference on Data Mining*. SIAM, 246–254.

[21] Sudeep Sarkar and Kim L Boyer. 1998. Quantitative measures of change based on feature organization: Eigenvalues and eigenvectors. *Computer vision and image understanding* 71, 1 (1998), 110–136.

[22] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. 2008. Collective classification in network data. *AI magazine* 29, 3 (2008), 93–93.

[23] Jianbo Shi and Jitendra Malik. 2000. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence* 22, 8 (2000), 888–905.

[24] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. 2019. Heterogeneous graph attention network. In *The world wide web conference*. 2022–2032.

[25] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. 2019. Simplifying graph convolutional networks. In *International conference on machine learning*. PMLR, 6861–6871.

[26] Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, and Edward Y. Chang. 2015. Network Representation Learning with Rich Text Information. In *IJCAI*.

[27] Chong You, Chun-Guang Li, Daniel P Robinson, and René Vidal. 2016. Oracle based active set algorithm for scalable elastic net subspace clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3928–3937.

[28] Chong You, Daniel Robinson, and René Vidal. 2016. Scalable sparse subspace clustering by orthogonal matching pursuit. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3918–3927.

[29] Shangzhi Zhang, Chong You, René Vidal, and Chun-Guang Li. 2021. Learning a self-expressive network for subspace clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12393–12403.

[30] Hao Zhu and Piotr Koniusz. 2021. Simple Spectral Graph Convolution. In *9th International Conference on Learning Representations, ICLR, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.