

# Subspace Co-clustering avec Convolution Bilatérale sur Graphe

Chakib Fettal<sup>\*,\*\*</sup>, Lazhar Labiod<sup>\*</sup>, Mohamed Nadif<sup>\*</sup>

<sup>\*</sup> Centre Borelli, UMR 9010  
Université Paris Cité

{prenom.nom}@u-paris.fr

<sup>\*\*</sup>Informatique Caisse des Dépôts et Consignations

**Résumé.** Le *subspace clustering* vise à partitionner un ensemble d'observations de haute dimension. Si cette approche a donné de bons résultats dans le domaine du partitionnement d'images, elle s'est avérée inefficace pour le partitionnement de données sparsées comme c'est le cas des données matrices termes-documents. Une extension appropriée de cette approche au co-clustering, particulièrement efficace sur des données sparsées, s'avère utile pour traiter de données attribuées. Ainsi, nous traitons le problème de la sparsité par le biais d'une convolution bilatérale sur graphe qui favorise l'effet de regroupement. Nous montrons la compétitivité de notre modèle par rapport à l'état de l'art sur des ensembles de données de graphes attribués en termes de performance et d'efficacité computationnelle.

## 1 Introduction et Contexte

Le présent article est un résumé de l'article publié dans la conférence CIKM (Fettal et al., 2022b). Le *subspace clustering* (Parsons et al., 2004) consiste à regrouper des observations (ou éléments) en fonction des sous-espaces qui les contiennent. Il existe une variété d'approches pour résoudre ce problème, dont beaucoup d'entre elles considèrent la formulation auto-expressive où l'on suppose que chaque élément peut être écrit comme une combinaison linéaire des éléments dans le même sous-espace. Typiquement, la formulation générique est donnée par

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

où  $\mathbf{X} \in \mathbb{R}^{n \times d}$ ,  $\mathbf{R} \in \mathbb{R}^{n \times n}$  est appelée la matrice d'auto-représentation,  $\Omega(\mathbf{R})$  sert de terme de régularisation pour induire des propriétés souhaitables sur  $\mathbf{R}$  et éviter les solutions triviales (telle que  $\mathbf{R} = \mathbf{I}$ ), et  $\mathcal{R}$  est la région réalisable.

Etant donné une solution optimale  $\mathbf{R}^*$ , une matrice d'affinité est générée sur la base des amplitudes des entrées de  $\mathbf{R}^*$ , en utilisant généralement  $|\mathbf{R}^* + \mathbf{R}^{*\top}|/2$ , et une partition des observations est ensuite générée en utilisant une méthode de partitionnement de graphes, par exemple l'algorithme de partitionnement spectral (Shi et Malik, 2000).

Les méthodes de type *subspace clustering* basées sur la propriété d'auto-expression (Zhang et al., 2021) ont été largement utilisées pour regrouper des ensembles de données de type image

en raison de l’hypothèse selon laquelle de telles données sont souvent tirées de multiples sous-espaces de faible dimension. L’une des premières approches était le subspace clustering par régression des moindres carrés (LSR) (Lu et al., 2012) qui tire parti d’un effet de regroupement basé sur la corrélation des données pour effectuer la segmentation. Des approches plus sophistiquées, qui constituent l’état de l’art en la matière, ont ensuite été proposées, telles que l’*Elastic-net Subspace Clustering* (EnSC) (You et al., 2016a) et le subspace clustering through *orthogonal matching pursuit* (SSC-OMP) (You et al., 2016b). Cependant, bien que les données textuelles répondent également à cette hypothèse, aucune approche de subspace clustering auto-expressive spécifiquement adaptée au texte n’a été proposée à notre connaissance. Cela peut peut-être s’expliquer par le fait que les ensembles de données de type termes-documents sont généralement beaucoup plus grands et plus creux que les ensembles de données d’images et que, par conséquent, peu de sous-espaces communs peuvent être identifiés.

Dans cet article, nous proposons un modèle de subspace clustering adapté aux matrices termes-documents par le biais du concept de co-clustering, c’est-à-dire en exploitant l’interaction entre les ensembles documents et termes pour générer une segmentation simultanée des deux. Nous proposons également un moyen de surmonter le problème éventuel de l’existence de peu de sous-espaces communs pour les documents/termes en utilisant une convolution de graphe à deux sens qui consiste en une étape de prétraitement de lissage laplacien pondéré inspirée par le réseau convolutif de graphe simple (Wu et al., 2019).

## 2 Méthode Proposée

**Notation** Les matrices sont désignées par des lettres majuscules en gras et les vecteurs par des lettres minuscules en gras. Étant donné une matrice  $\mathbf{X}$ , sa  $i$ -ième ligne est désignée par  $\mathbf{x}_i$  et sa  $j$ -ième colonne par  $\mathbf{x}'_j$ .  $\mathbf{I}_n$  est la matrice identité de taille  $n \times n$ . La norme de Frobenius est désignée par  $\|\cdot\|$ ,  $k$  et  $g$  désignent respectivement le nombre de classes des documents et des termes. La fonction  $[\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}] = \text{SVD}(\mathbf{X})$  donne la décomposition en valeurs singulières de la matrice  $\mathbf{X}$  où  $\mathbf{U}$  et  $\mathbf{V}$  sont les vecteurs singuliers gauche et droit, et  $\mathbf{\Sigma}$  est la matrice diagonale contenant les valeurs singulières triées par ordre décroissant.

### 2.1 Subspace Co-clustering Auto-expressif

L’avantage du co-clustering (Govaert et Nadif, 2013; Salah et Nadif, 2017, 2019; Affeldt et al., 2020, 2021; Riverain et al., 2022) est qu’il utilise la dualité inhérente entre les lignes et les colonnes des tableaux de données, ce qui peut conduire à une amélioration du partitionnement selon les deux dimensions. Par exemple, dans le cas des matrices de termes-documents, le co-clustering s’appuie sur l’ensemble des termes pour effectuer le partitionnement des documents et vice-versa. Motivés par cette observation, nous formulons le problème du subspace co-clustering comme suit. Étant donné une matrice de termes-documents  $\mathbf{X} \in \mathbb{R}_+^{n \times d}$ , nous proposons d’optimiser

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

où  $\mathbf{R} \in \mathbb{R}^{n \times n}$  et  $\mathbf{C} \in \mathbb{R}^{d \times d}$  sont respectivement les matrices d’auto-représentation des documents et des termes,  $\mathcal{R}$  et  $\mathcal{C}$  sont les régions réalisables,  $\Omega(\mathbf{R}, \mathbf{C})$  est le terme de régulari-

sation où la régularisation de  $\mathbf{R}$  et de  $\mathbf{C}$  peut être indépendante, c'est-à-dire que  $\Omega(\mathbf{R}, \mathbf{C}) = \Omega_{\mathbf{R}}(\mathbf{R}) + \Omega_{\mathbf{C}}(\mathbf{C})$ , ou dépendante.

## 2.2 Promouvoir l'effet de regroupement par une convolution bilatérale

Certaines méthodes de subspace clustering (Lu et al., 2012; Hu et al., 2014; Lu et al., 2013; Diallo et al., 2021) attribuent la performance de leur clustering à l'effet de regroupement.

**Definition (effet de regroupement)** : Étant donné une matrice de données  $\mathbf{X}$ , une matrice d'auto-représentation  $\mathbf{R}$  a un effet de regroupement si

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

Cela peut poser un problème dans le cas d'un texte en raison de sa haute dimensionnalité et de sa sparsité, car les observations (documents ou termes) peuvent ne pas être suffisamment "proches" au sens de la propriété d'auto-expression pour être regroupées de manière significative. Cela signifie que les observations ont besoin d'une sorte de lissage pour aider les algorithmes de subspace clustering à trouver des sous-espaces communs. Nous proposons de résoudre ce problème par une convolution de graphe à deux voies. Cela nécessite deux matrices de similarités qui agiront comme des graphes sur les lignes  $\mathbf{S}_{\mathbf{R}}$  et les colonnes  $\mathbf{S}_{\mathbf{C}}$ . Ces matrices peuvent être construites par une mesure de similarité sur les données ou être fournies a priori, par exemple dans le cas de graphes attribués.

Une intuition peut être tirée du fait que les lignes et les colonnes de  $\mathbf{S}_{\mathbf{R}}^p \mathbf{X} \mathbf{S}_{\mathbf{C}}^q$ , à mesure que les ordres de propagation  $p, q$  augmentent, deviennent plus lisses en étant moyennées jusqu'à leurs  $p$ -ième et  $q$ -ième voisins respectivement, à la manière d'un lissage laplacien. Cette opération rend donc les documents et les termes de plus en plus similaires. Grâce à la propriété de partitionnement, cela implique que les vecteurs d'auto-représentation devraient également devenir plus similaires, ce qui conduit à un partitionnement plus significatif.

Le défi consiste à choisir des ordres de propagation appropriés, car des valeurs élevées peuvent entraîner un lissage excessif et faire en sorte que toutes les observations se ressemblent. Notre problème devient

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

Dans ce qui suit, nous ferons référence à la matrice lissée  $\mathbf{S}_{\mathbf{R}}^p \mathbf{X} \mathbf{S}_{\mathbf{C}}^q$  en utilisant  $\mathbf{H}$  puisque cette opération peut être considérée comme une sorte d'étape de prétraitement des données, indépendante du modèle de clustering. Notez que la complexité de cette opération est en  $\mathcal{O}(p\|\mathbf{S}_{\mathbf{R}}\|_0 + q\|\mathbf{S}_{\mathbf{C}}\|_0)$  où  $\|\cdot\|_0$  est la norme zéro qui donne le nombre d'entrées non nulles de son argument.

## 2.3 Contraintes d'orthogonalité

Pour résoudre le problème de la complexité, nous proposons d'introduire les contraintes suivantes :  $\mathbf{R} = \mathbf{Z}\mathbf{Z}^{\top}$  et  $\mathbf{C} = \mathbf{W}\mathbf{W}^{\top}$  où  $\mathbf{Z} \in \mathbb{R}^{n \times k}$  et  $\mathbf{W} \in \mathbb{R}^{d \times g}$  sont semi-orthogonales, c'est-à-dire que  $\mathbf{Z}^{\top}\mathbf{Z} = \mathbf{I}_k$  et  $\mathbf{W}^{\top}\mathbf{W} = \mathbf{I}_g$ . La nouvelle formulation du problème est

$$\min_{\mathbf{Z}, \mathbf{W}} \|\mathbf{H} - \mathbf{Z}\mathbf{Z}^{\top}\mathbf{H}\mathbf{W}\mathbf{W}^{\top}\|^2 \text{ s.t. } \mathbf{Z}^{\top}\mathbf{Z} = \mathbf{I}_k, \mathbf{W}^{\top}\mathbf{W} = \mathbf{I}_g \quad (4)$$

À première vue, ce problème nécessite également un schéma de résolution alternatif utilisant deux règles de mise à jour que nous obtenons en fixant  $\mathbf{W}$  et en résolvant pour  $\mathbf{Z}$  et vice versa

$$\begin{aligned} \mathbf{Z} &= [\mathbf{u}'_1, \dots, \mathbf{u}'_k] \quad \text{s.t.} \quad [\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}] = \text{SVD}(\mathbf{H}\mathbf{W}) \\ \mathbf{W} &= [\mathbf{u}'_1, \dots, \mathbf{u}'_k] \quad \text{s.t.} \quad [\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}] = \text{SVD}(\mathbf{H}^\top \mathbf{Z}). \end{aligned} \quad (5)$$

**Calcul Rapide de  $\mathbf{Z}^*$  et  $\mathbf{W}^*$ .** Le problème précédent peut être résolu efficacement en utilisant une seule SVD tronquée. Ceci est une conséquence de la proposition suivante.

**Proposition 1.** Le processus alternatif défini dans le système de (5) converge vers  $\mathbf{Z}$  et  $\mathbf{W}$  étant les vecteurs singuliers tronqués gauche et droite de  $\mathbf{H}$  respectivement.

L'algorithme est donc plus efficace puisque nous contournerons l'étape itérative. Cependant, l'interaction entre les lignes et les colonnes reste implicite puisque la solution résultante est également une solution au problème d'optimisation alternée susmentionné où l'interaction est explicite.

*Démonstration.* Supposons, sans perte de généralité, que  $k \leq \text{rank}(\mathbf{H})$ . Nous avons  $k = \text{rank}(\mathbf{Z}) = \text{rank}(\mathbf{W})$  impliquant  $\text{rank}(\mathbf{Z}\mathbf{Z}^\top \mathbf{H}\mathbf{W}\mathbf{W}^\top) \leq k$ . Cela signifie que nous recherchons la meilleure approximation de rang  $k$ . Étant donné  $[\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}] = \text{SVD}(\mathbf{H})$ , en fixant  $\mathbf{Z} = \mathbf{U}_k = [\mathbf{u}'_1, \dots, \mathbf{u}'_k]$  et  $\mathbf{W} = \mathbf{V}_k = [\mathbf{v}'_1, \dots, \mathbf{v}'_k]$ . Nous avons

$$\|\mathbf{H} - \mathbf{Z}\mathbf{Z}^\top \mathbf{H}\mathbf{W}\mathbf{W}^\top\|^2 = \|\mathbf{H} - \mathbf{U}_k \mathbf{U}_k^\top \mathbf{U} \mathbf{\Sigma} \mathbf{V}^\top \mathbf{V}_k \mathbf{V}_k^\top\|^2 = \|\mathbf{H} - \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^\top\|^2 \quad (6)$$

qui, selon le théorème d'Eckart-Young-Mirsky, est la valeur optimale du problème d'approximation du rang  $k$  de  $\mathbf{H}$ .  $\square$

A partir du résultat précédent, nous pouvons montrer que notre approche a un effet de regroupement approximatif.

**Proposition 2.** Étant donné la matrice  $\mathbf{H}$ , les solutions  $\mathbf{R}$  et  $\mathbf{C}$  présentent un effet de regroupement sur la matrice  $\tilde{\mathbf{H}}$ , la meilleure approximation rang- $k$  de la somme puisque  $\mathbf{z}_i = \mathbf{\Sigma}_k^{-1} \mathbf{W}^\top \tilde{\mathbf{h}}_i$  et  $\mathbf{w}_i = \mathbf{\Sigma}_k^{-1} \mathbf{Z}^\top \tilde{\mathbf{h}}_i$ .

**Partitionnement Spectrale Rapide de  $\mathbf{R}^*$  and  $\mathbf{C}^*$ .** La matrice de coefficients optimale  $\mathbf{R}^* = \mathbf{Z}^* \mathbf{Z}^{*\top}$  est symétrique par construction, cependant ses entrées ne sont pas nécessairement négatives ce qui implique de devoir utiliser la valeur absolue par élément pour obtenir une matrice d'affinité valide. Cela détruirait toutes les informations que nous avons déjà sur la décomposition de  $\mathbf{R}^*$  en  $\mathbf{Z}^* \mathbf{Z}^{*\top}$  puisque généralement il n'y a aucune relation entre le spectre d'une matrice et son spectre après application d'une fonction par entrée. Nous contournerons ce problème en considérant plutôt la matrice d'affinité  $\mathbf{K}_R = (r_{ij} + 1)_{ij}^2$ . Nous avons alors  $\mathbf{K}_R = \langle \varphi(\mathbf{Z}^*), \varphi(\mathbf{Z}^*) \rangle$  où  $\varphi$  est une feature map pour le noyau polynomial du second degré  $\mathbf{K}_R$  appliqué sur les vecteurs de lignes de  $\mathbf{Z}$ , c'est-à-dire

$$\begin{aligned} \varphi(\mathbf{z}) &= \langle z_k^2, \dots, z_1^2, \sqrt{2}z_k z_{k-1}, \dots, \sqrt{2}z_k z_1, \sqrt{2}z_{k-1} z_{k-2}, \\ &\quad \dots, \sqrt{2}z_{k-1} z_1, \dots, \sqrt{2}z_2 z_1, \sqrt{2}z_k, \dots, \sqrt{2}z_1, 1 \rangle \end{aligned} \quad (7)$$

avec  $\varphi : \mathbb{R}^k \rightarrow \mathbb{R}^{\binom{k+2}{2}}$ . Toute carte de caractéristiques pour un noyau non négatif à l'entrée est une alternative possible. Nous avons choisi ici la carte de caractéristiques exacte la plus simple possible puisque la transformation n'entraîne pas une augmentation trop importante de la dimensionnalité des entrées lorsque  $k \ll n, d$ . Des approximations peuvent être utilisées afin de travailler avec des noyaux de carte de caractéristiques de dimension infinie, par exemple le noyau RBF. Nous proposons ensuite d'effectuer le clustering spectral directement sur la matrice d'affinité au lieu du Laplacien graphique comme dans (Sarkar et Boyer, 1998). Comme les vecteurs propres de  $\mathbf{K}_R$  sont les mêmes que les vecteurs singuliers gauches de  $\varphi(\mathbf{Z}^*)$ , le processus est beaucoup plus rapide puisque  $k \ll d$ . Pour obtenir un clustering à partir de  $\mathbf{C}^*$ , les opérations sont les mêmes. La complexité globale du calcul est alors de  $\mathcal{O}((nd + nk^2 + dk^2) \log(k))$  alors que la complexité spatiale est de  $\mathcal{O}(nk^2 + dk^2)$ .

### 3 Expériences

#### 3.1 Configuration Expérimentale

**Données.** Nous utilisons quatre réseaux de citation de graphes attribués. Les statistiques sommaires sont disponibles dans le tableau 1.

**Modèles Comparatifs.** Nous comparons notre modèle à des modèles de clustering et de co-clustering qui utilisent uniquement la matrice de caractéristiques des nœuds d'entrée  $\mathbf{X}$  ou qui utilisent à la fois  $\mathbf{A}$  et  $\mathbf{X}$ , c'est-à-dire des modèles de clustering/co-clustering de graphes attribués. L'algorithme  $k$ -means est notre référence. Nous utilisons également les modèles de sous-espace clustering LSR, EnSC et SSC-OMP susmentionnés. Pour les modèles de clustering de graphes attribués, nous utilisons GIC (Mavromatis et Karypis, 2021), S<sup>2</sup>GC (Zhu et Koniusz, 2021), et GCC (Fettal et al., 2022a) qui propose un schéma simultané d'apprentissage de représentation et de partitionnement des nœuds. Nous comparons aussi le modèle à l'algorithme de co-clustering spectral (Dhillon, 2001). Pour les Modèles de co-clustering à graphes attribués, le seul modèle de ce type est le CFOND (Guo et al., 2018).

TAB. 1 – Description des données.

Nom	#Nœuds	#Arêtes	#Features	#Classes
ACM	3025	9150593	1870	3
CiteSeer	3327	4732	3703	6
PubMed	19717	44338	500	3
Wiki	2405	17981	4973	17

**Paramètres Expérimentaux.** Pour notre méthode, nous utilisons comme graphe de lignes, la matrice d'adjacence fournie dans les jeux de données :  $\mathbf{S}_R = \mathbf{A}$ . Pour les colonnes, nous utilisons  $\mathbf{S}_C = \left( \max \left\{ \log \left( \frac{c_{i,j}}{c_{i,i} c_{j,j}} c_{ij} \right), 0 \right\} \right)_{ij}$  où  $\mathbf{C} = \mathbf{X}^\top \mathbf{X}$ , qui est la matrice d'information mutuelle ponctuelle non négative (Church et Hanks, 1990) des termes ; Intuitivement  $s_{C_{ij}}$  donne la parenté sémantique du terme  $i$  et  $j$ , plus la valeur est grande, plus ces termes sont liés. Les deux matrices sont ensuite ajoutées aux auto-boucles et normalisées comme dans (Fettal et al., 2022a). L'ordre de propagation en ligne  $p$  est sélectionné en utilisant la règle de sélection proposée dans (Fettal et al., 2022a), tandis que pour celui en colonne nous fixons  $q = 1$ . Nous effectuons dix exécutions pour chaque modèle. Nous utilisons l'implémentation

TAB. 2 – Performance de clustering sur les ensembles de données, moyenne sur dix exécutions. Les meilleurs résultats sont mis en évidence en gras. Notre modèle est compétitif par rapport à l'état de l'art car il obtient les meilleurs résultats sur la plupart des jeux de données tout en ayant de faibles écarts types.

Method	Input	ACM			CiteSeer			PubMed			Wiki		
		Acc	NMI	ARI									
<i>k</i> -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	<b>33.3±1.5</b>
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
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 <sup>2</sup> GC	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	<b>32.5±0.0</b>	<b>33.2±0.0</b>	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	<b>45.0±0.2</b>	<b>45.4±0.1</b>	70.8±0.0	32.3±0.0	<b>33.2±0.0</b>	54.1±0.8	<b>55.0±0.2</b>	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	<b>81.4±0.0</b>	<b>50.0±0.0</b>	<b>53.9±0.0</b>	<b>69.5±0.0</b>	43.5±0.0	43.7±0.0	<b>70.9±0.0</b>	31.7±0.0	<b>33.2±0.0</b>	<b>59.3±0.6</b>	53.9±0.9	32.7±1.4

et les paramètres fournis par les auteurs lorsqu'ils sont disponibles. Pour les modèles qui utilisent un paramètre  $p$  comme le nôtre, nous exécutons la règle de sélection proposée jusqu'à convergence sans spécifier de maximum  $p$ , par souci d'équité. Toutes les expériences ont été réalisées sur la même machine.

### 3.2 Partitionnement des documents

Nous comparons les méthodes de partitionnement de documents en utilisant les métriques Clustering Accuracy (Acc), Normalized Mutual Information (NMI) et Adjusted Rand Score (ARI). Le tableau 2 montre les performances des différents modèles. Les méthodes qui utilisent à la fois la structure et les caractéristiques du graphe sont plus performantes que celles qui n'utilisent que les caractéristiques des nœuds, à l'exception de l'ACM où le graphe n'est pas informatif (la plupart des entrées sont égales à un).

Nous avons utilisé ce jeu de données pour montrer la robustesse de notre modèle face à une structure de graphe non informative, par rapport aux modèles de partitionnement de graphes attribués à l'état de l'art. Nous constatons que notre approche est compétitive et surpasse les autres modèles sur tous les jeux de données en termes de précision. Elle présente également un écart-type proche de zéro pour la plupart des mesures, ce qui est un signe de robustesse. Nous reportons dans le tableau 3 les temps d'exécution de notre algorithme par rapport aux autres modèles de subspace clustering. Nous pouvons voir que notre approche de subspace clustering est plus rapide dans les différents jeux de données par des marges significatives, même si elle génère un clustering des documents et des termes.

TAB. 3 – Temps d'apprentissage en secondes des différents modèles de subspace clustering, avec une moyenne de dix essais.

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	<b>9.7</b>	<b>7.4</b>	<b>28.9</b>	<b>8.8</b>

## 4 Conclusion

Nous avons proposé SCC, une nouvelle approche pour tirer parti du subspace clustering pour les données textuelles par le biais du co-clustering et de la convolution de graphe à deux voies. Celle-ci contourne les problèmes de complexité spatiale et de calcul du subspace clustering en utilisant des matrices de facteurs et des cartes de caractéristiques à noyau non négatif. Les expériences ont montré que notre modèle est compétitif par rapport à l'état de l'art pour le partitionnement de nœuds de graphes attribués en termes de performance et de robustesse. Une version de ce travail dédiée au clustering a été récemment proposée dans (Fettal et al., 2023).

**Remerciements.** Ce travail a été financé par la Caisse des Dépôts et Consignations (CDC), l'ANRT et l'Idex-Spectrans d'Université Paris Cité.

## Références

- Affeldt, S., L. Labiod, et M. Nadif (2020). Ensemble block co-clustering : a unified framework for text data. In *CIKM*, pp. 5–14.
- Affeldt, S., L. Labiod, et M. Nadif (2021). Regularized dual-PPMI co-clustering for text data. In *SIGIR*, pp. 2263–2267.
- Church, K. et P. Hanks (1990). Word association norms, mutual information, and lexicography. *Computational linguistics* 16(1), 22–29.
- Dhillon, I. S. (2001). Co-clustering documents and words using bipartite spectral graph partitioning. In *SIGKDD*, pp. 269–274.
- Diallo, A. W., N. Niang, et M. Ouattara (2021). Sparse subspace k-means. In *2021 International Conference on Data Mining Workshops (ICDMW)*, pp. 678–685.
- Fettal, C., L. Labiod, et M. Nadif (2022a). Efficient graph convolution for joint node representation learning and clustering. In *WSDM*, pp. 289–297.
- Fettal, C., L. Labiod, et M. Nadif (2022b). Subspace co-clustering with two-way graph convolution. In *CIKM*, pp. 3938–3942.
- Fettal, C., L. Labiod, et M. Nadif (2023). Scalable attributed-graph clustering. In *AAAI 2023*.
- Govaert, G. et M. Nadif (2013). *Co-clustering : models, algorithms and applications*. John Wiley & Sons.
- Guo, T., S. Pan, X. Zhu, et C. Zhang (2018). Cfond : consensus factorization for co-clustering networked data. *IEEE Transactions on Knowledge and Data Engineering* 31(4), 706–719.
- Hu, H., Z. Lin, J. Feng, et J. Zhou (2014). Smooth representation clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3834–3841.
- Lu, C., J. Feng, Z. Lin, et S. Yan (2013). Correlation adaptive subspace segmentation by trace lasso. In *ICV*, pp. 1345–1352.
- Lu, C.-Y., H. Min, Z.-Q. Zhao, L. Zhu, D.-S. Huang, et S. Yan (2012). Robust and efficient subspace segmentation via least squares regression. In *ECCV*, pp. 347–360. Springer.
- Mavromatis, C. et G. Karypis (2021). Graph infoclust : Maximizing coarse-grain mutual information in graphs. In *PAKDD (1)*, pp. 541–553.

- Parsons, L., E. Haque, et H. Liu (2004). Subspace clustering for high dimensional data : a review. *Acm sigkdd explorations newsletter* 6(1), 90–105.
- Riverain, P., S. Fossier, et M. Nadif (2022). Semi-supervised latent block model with pairwise constraints. *Machine Learning* 111(5), 1739–1764.
- Salah, A. et M. Nadif (2017). Model-based von mises-fisher co-clustering with a conscience. In *SDM*, pp. 246–254. SIAM.
- Salah, A. et M. Nadif (2019). Directional co-clustering. *Advances in Data Analysis and Classification* 13(3), 591–620.
- Sarkar, S. et K. L. Boyer (1998). Quantitative measures of change based on feature organization : Eigenvalues and eigenvectors. *Computer vision and image understanding* 71(1), 110–136.
- Shi, J. et J. Malik (2000). Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence* 22(8), 888–905.
- Wu, F., A. Souza, T. Zhang, C. Fifty, T. Yu, et K. Weinberger (2019). Simplifying graph convolutional networks. In *ICML*, pp. 6861–6871. PMLR.
- You, C., C.-G. Li, D. P. Robinson, et R. Vidal (2016a). Oracle based active set algorithm for scalable elastic net subspace clustering. In *ICVP*, pp. 3928–3937.
- You, C., D. Robinson, et R. Vidal (2016b). Scalable sparse subspace clustering by orthogonal matching pursuit. In *ICVP*, pp. 3918–3927.
- Zhang, S., C. You, R. Vidal, et C.-G. Li (2021). Learning a self-expressive network for subspace clustering. In *IEEE/CVF*, pp. 12393–12403.
- Zhu, H. et P. Koniusz (2021). Simple spectral graph convolution. In *9th International Conference on Learning Representations, ICLR, Virtual Event, Austria, May 3-7, 2021*.

## Summary

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.