Ensemble Block Co-clustering: A Unified Framework for Text Data

Séverine Affeldt, Lazhar Labiod, Mohamed Nadif

Ensemble Block Co-clustering: A Unified Framework for Text Data

Séverine Affeldt
Université de Paris, LIPADE
F-75006 Paris
severine.affeldt@u-paris.fr

Lazhar Labiod
Université de Paris, LIPADE
F-75006 Paris
lazhar.labiod@u-paris.fr

Mohamed Nadif
Université de Paris, LIPADE
F-75006 Paris
mohamed.nadif@u-paris.fr

ABSTRACT

In this paper, we propose a unified framework for Ensemble Block Co-clustering (EBCO), which aims to fuse multiple *basic* co-clusterings into a consensus structured affinity matrix. Each co-clustering to be fused is obtained by applying a co-clustering method on the same document-term dataset. This fusion process reinforces the individual quality of the multiple basic data co-clusterings within a single consensus matrix. Besides, the proposed framework enables a completely unsupervised co-clustering where the number of co-clusters is automatically inferred based on the non trivial generalized modularity. We first define an explicit objective function which allows the joint learning of the basic co-clusterings aggregation and the consensus block co-clustering. Then, we show that EBCO generalizes the one side ensemble clustering to an ensemble block co-clustering context. We also establish theoretical equivalence to spectral co-clustering and weighted double spherical k-means clustering for textual data. Experimental results on various real-world document-term datasets demonstrate that EBCO is an efficient competitor to some state-of-the-art ensemble and co-clustering methods.

CCS CONCEPTS

• **Computing methodologies** → **Unsupervised learning**; **Ensemble methods**; **Cluster analysis**; • **Information systems** → *Clustering and classification*; • **Mathematics of computing** → Probability and statistics.

KEYWORDS

co-clustering; ensemble method; information retrieval; text mining

ACM Reference Format:
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 and Knowledge Management (CIKM '20), October 19–23, 2020, Ireland*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3340531.3412058

CIKM '20, October 19–23, 2020, Ireland

ACM ISBN 978-1-4503-6859-9/20/10…$15.00
https://doi.org/10.1145/3340531.3412058

1 INTRODUCTION

Clustering is the process of organizing similar objects into meaningful clusters. This approach is essential in many fields, including data science, machine learning, information retrieval, bio-informatics and computer vision, to deal with massive data. The clustering problem has been extensively addressed by different communities and many different approaches have been developed from various perspectives with various focuses. Although the purpose of clustering may seem simple, it is however an inherently difficult problem; different clustering algorithms and even multiple trials of the same algorithm may produce different results due to the fact that the initialization is not deterministic. To overcome the resulting instability and improve clustering performance, the *ensemble clustering* approach became an interesting alternative and therefore emerged as an important extension of the classical clustering problem. It refers to the following problem: given a number of different (input) clusterings that have been generated from a dataset, find a single final (consensus) clustering that is a better fit in some sense than the existing clusterings [28].

Ensemble clustering has been extensively studied and many different approaches have been developed from various perspectives with various focuses [3, 9, 19, 30, 33–35, 38]. The main contribution of the co-association method is the redefinition of the consensus clustering problem as a classical graph partition problem. Thereby, Strehl and Ghosh [28] developed three graph-based algorithms for consensus clustering. Fred and Jain [10] applied the agglomerative hierarchical clustering. In addition, there are some other interesting approaches for the ensemble clustering, such as the ones based on Information theoretic method [30], EM algorithm with a finite mixture of multinomial distributions [31], Matrix factorization based method [18, 19, 35], and kernel-based methods [33], respectively. Recently, Liu *et al.* [20] proposed a spectral ensemble clustering method, which ran spectral clustering on the co-association matrix and transformed it as a weighted k-means problem to achieve high efficiency. Tao *et al.* [29] proposed a unified framework to simultaneously learn a robust representation for the co-association matrix and find the final consensus partition. However, existing ensemble techniques are primarily designed for one side clustering methods, and few research efforts have been reported for ensemble co-clustering methods. For datasets arising in text mining and bioinformatics where the data is represented in a very high dimensional space, clustering both dimensions of data matrix simultaneously is often more desirable than traditional one side clustering. Co-clustering [1, 11, 17, 26] which is a simultaneous clustering of objects and features of data matrix consists generally in interlacing object clusterings with feature clusterings at each iteration; co-clustering exploits the duality between objects and

features which allows to effectively deal with high dimensional data. In this way, co-clustering algorithms aim to reorganize the initial data matrix into homogeneous co-clusters[1] or biclusters, that can therefore be seen as subsets characterized by a set of observations and a set of features. Furthermore, co-clustering implicitly performs an adaptive dimensionality reduction at each iteration, leading to better object clustering accuracy compared to one side clustering methods.

The success on one hand of ensemble methods in classification and clustering, and on the other hand of co-clustering techniques to deal with high dimensional and sparse data, provides the main motivations for applying ensemble methods in document-term matrix co-clustering. In this work, we propose a novel Ensemble Block Co-clustering (EBCO) framework in which the input is a collection of document-term matrix co-clusterings. The output of the framework is a consensus block co-clustering. The Figure 1 illustrates the detailed conceptual framework of the EBCO method.

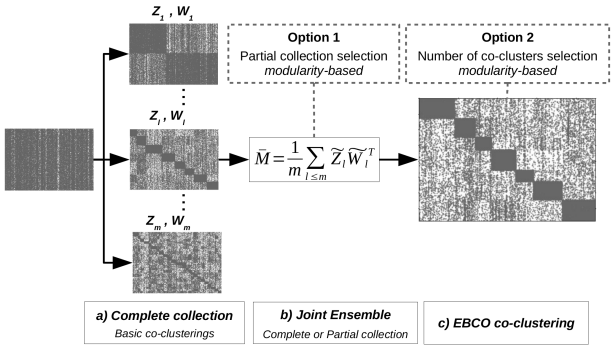


Figure 1: EBCO Framework: *a*) **From a document-term matrix, let be a collection of** m **co-clusterings or co-partitions** $(\mathbf{Z}_l, \mathbf{W}_l)$**;** $l = 1, \ldots, m$ **obtained by a given co-clustering algorithm.** *b*) **Construct a combined affinity matrix which integrates information from all basic co-clusterings by** $\overline{\mathbf{M}} = \frac{1}{m}\sum_{l=1}^{m} \mathbf{M}_l$ **where** $\mathbf{M}_l = \mathbf{Z}_l\mathbf{W}_l^\top$**.** *c*) **Factorize the matrix** $\overline{\mathbf{M}}$ **into a common consensus block co-clustering** $\mathbf{Q} = \mathbf{Z}\mathbf{W}^\top$**.**

This paper provides a unified view on combining multiple co-clusterings by studying connections among various consensus and co-clustering criteria. The contribution of this paper is four-fold:

- We first tackle the problem of combining multiple co-clusterings as a bipartite graph co-clustering problem from a new perspective.
- Then, we propose an explicit objective function to generate and supervise the basic co-clusterings, and to determine the final co-clustering.
- We show the connections between various consensus and co-clustering criteria. In particular, we demonstrate that the proposed method employs a generalized co-association matrix to find the consensus co-partition and equivalently results in a weighted k-means co-clustering, which decreases the time and space complexity.
- Finally we propose a method to assess the number of co-clusters. To the best of our knowledge, this is first time an ensemble co-clustering is considered while estimating the number of co-clusters.

The rest of the article is organized as follows: Section 2 discusses the related work; Section 3 details the EBCO framework for ensemble co-clustering; Section 4 investigates gateways and connections to some state-of-the-art consensus and co-clustering techniques for textual data. Section 5 shows experimental evaluations and results analysis; and, finally, Section 6 concludes the paper and discusses future work.

2 RELATED WORK

Co-clustering, under various names, has been successfully used in a wide range of application domains where the co-clusters can take different forms [32]. If it has become popular, it is mainly through its numerous applications in different domains. For instance, in bioinformatics co-clustering, referred to as biclustering [21], is used to cluster genes and experimental conditions simultaneously. In collaborative filtering [5], it is used to group users and items simultaneously, and in text mining [2, 12] to group terms and documents simultaneously. If there are many works on ensemble clustering, few of them are devoted to co-clustering. To improve the performance of popular biclustering methods devoted to microarrays data, in [13] the authors proposed to adapt the approach of bagging to biclustering problems and in [14] they formalized the ensemble biclustering through a problem of binary triclustering. However, by contrast with our proposal, these approaches are unfortunately not appropriate for textual data due the nature of the data and the objective of the biclustering algorithms considered.

On the other hand, several approaches, inspired by the spectral clustering principle leading to a subspace on which k-means is applied, have been proposed. For instance, Huang *et al.* [15] designed a spectral co-clustering ensemble algorithm (SCCE) formulated as a bipartite graph partition problem. More specifically, let $\mathbf{X}$ be a data matrix (n objects described by d features). A SVD is performed on the $(n + d) \times (n + d)$ adjacency matrix, then k-means is applied on the obtained subspace. More recently, Xianxue *et al.* [39] proposed a co-clustering ensemble algorithm (CoCE) which, unlike SCCE, first evaluates the quality of the base co-clusters by measuring feature-to-object relevance, and then the consensus process relies on the resulting hybrid graph. To obtain the object and feature clusters, k-means is performed separately on the obtained subspace by performing a SVD on a small graph Laplacian matrix. In fact, CoCE is formulated as a trace minimization problem and introduces a block-wise matrix multiplication technique to perform the optimization. The adjacency matrix is reduced from an $(n + d)$ square matrix to an n by d matrix. As a result, CoCE is distinct from SCCE and has a lower time complexity. Yet, the CoCE time complexity still remains considerably high, in particular when dealing with text data, that are generally sparse and high dimensional.

As we will see in detail, EBCO differs from the above-mentioned approaches at three stages; *1)* Our consensus approach is based on multiple co-clusterings while remaining in the spirit of a double k-means-like algorithm. It has therefore a considerably lower computational complexity. *2)* Our approach focuses on text data and

[1]Each co-cluster determines a submatrix of the original data matrix with some desired properties.

takes into account the nature of the data which exhibits *directional* characteristics. *3)* The evaluation of the number of co-clusters is addressed in our proposal unlike in all the approaches cited above.

3 ENSEMBLE BLOCK CO-CLUSTERING (EBCO)

As we focus on document-term matrices, let $\mathbf{X} = (x_{ij})$ be a data matrix of size $n \times d$, where $x_{ij} \in \mathbb{N}$ denotes the frequency of term j in document i. The i^{th} row (document) of this matrix is represented by a vector $\mathbf{x}_i = (x_{i1}, \ldots, x_{id})^\top$, where $\top$ denotes the transpose. The partition of the set of documents I into g clusters can be represented by a classification matrix $\mathbf{Z} = (z_{ik}) \in \{0,1\}^{n \times g}$ satisfying $\forall i, \sum_{k=1}^{g} z_{ik} = 1$. In the same way, we adopt the same notation for the partition of the set of terms J by considering a classification matrix $\mathbf{W} = (w_{jk}) \in \{0,1\}^{d \times g}$ satisfying $\forall j, \sum_{k=1}^{g} w_{jk} = 1$.

3.1 Problem definition

The techniques related to seriation aim to reorganize I and J according to a diagonal block correspondence [22]. This objective, called the block seriation problem, can be addressed by finding optimal partitions described by $\mathbf{Z}$ and $\mathbf{W}$ of I and J respectively. This task can be carried out by a co-clustering method or relying on a block seriation relation $\mathbf{Q} = (\mathbf{q}_{ij})$ defined on $I \times J$ by $\mathbf{Q} = \mathbf{Z}\mathbf{W}^\top$ where $\mathbf{q}_{ij} = 1$ if document i is in the same block as attribute j and $\mathbf{q}_{ij} = 0$ otherwise. Then,

$$\mathbf{q}_{ij} = \sum_{k=1}^{g} \mathbf{z}_{ik}\mathbf{w}_{jk} = (\mathbf{Z}\mathbf{W}^\top)_{ij}. \tag{1}$$

The matrix $\mathbf{Q}$ represents a block seriation relation (see [22] for further details), then it must respect the following properties,

- **Binarity.** $\mathbf{q}_{ij} \in \{0,1\}, \forall (i,j) \in I \times J$.
- **Assignment constraints.** These constraints ensure the bijective correspondence between classes of two partitions, meaning that each class of the partition of I has one and one corresponding class of the partition J, and conversely, these constraints are expressed linearly as follows,

$$\begin{cases} \sum_{j \in J} \mathbf{q}_{ij} \geq 1 & \forall i \in I \\ \sum_{i \in I} \mathbf{q}_{ij} \geq 1 & \forall j \in J. \end{cases}$$

- **Triad impossible.** The role of these constraints is to ensure the blocks disjoint structure which is expressed by the following system inequality,

$$\begin{cases} \mathbf{q}_{ij} + \mathbf{q}_{ij'} + \mathbf{q}_{i'j'} - \mathbf{q}_{i'j} - 1 \leq 1 \\ \mathbf{q}_{i'j'} + \mathbf{q}_{i'j} + \mathbf{q}_{ij} - \mathbf{q}_{ij'} - 1 \leq 1 \\ \mathbf{q}_{i'j} + \mathbf{q}_{ij} + \mathbf{q}_{ij'} - \mathbf{q}_{i'j'} - 1 \leq 1 \\ \mathbf{q}_{ij'} + \mathbf{q}_{i'j'} + \mathbf{q}_{i'j} - \mathbf{q}_{ij} - 1 \leq 1. \end{cases}$$

Furthermore, note that these constraints generalize the transitivity for non symmetric data. In the case where $I = J$, it is easy to show that the block seriation relation $\mathbf{Q}$ becomes an equivalence relation, i.e. $\mathbf{Q} = \mathbf{Z}\mathbf{Z}^\top$ or $\mathbf{Q} = \mathbf{W}\mathbf{W}^\top$.

Notice that the block seriation defined in Eq. (1) is not balanced by the row and column cluster size, meaning that a cluster might become small when affected by outliers. Thus we propose a new scaled block seriation relation that considers both row and column cluster sizes as follows,

$$\widetilde{\mathbf{q}}_{ij} = \sum_{k=1}^{g} \frac{\mathbf{z}_{ik}\mathbf{w}_{jk}}{\sqrt{\mathbf{z}_{.k}\mathbf{w}_{.k}}} = \sum_{k=1}^{g} \tilde{\mathbf{z}}_{ik}\tilde{\mathbf{w}}_{jk} = (\widetilde{\mathbf{Z}}\widetilde{\mathbf{W}}^T)_{ij} \tag{2}$$

where the cluster sizes of $\mathbf{Z}$ and $\mathbf{W}$ are on the diagonal of $\mathbf{D}_\mathbf{z} = \mathbf{Z}^\top\mathbf{Z}$ and $\mathbf{D}_\mathbf{w} = \mathbf{W}^\top\mathbf{W}$, respectively. Thereby we have $\widetilde{\mathbf{Z}} = \mathbf{Z}\mathbf{D}_\mathbf{z}^{-0.5}$ and $\widetilde{\mathbf{W}} = \mathbf{W}\mathbf{D}_\mathbf{w}^{-0.5}$.

3.2 Objective function

The problem of combining multiple co-clusterings can be described as follows. Let $\mathcal{M} = \{(\mathbf{Z}_l, \mathbf{W}_l); l = 1, ..., m\}$ be a set of m co-partitions[2] obtained by a given co-clustering algorithm. Each co-cluster can be modeled as a scaled block seriation relation in this way $\widetilde{\mathbf{M}}_l = \widetilde{\mathbf{Z}}_l\widetilde{\mathbf{W}}_l^\top, l = 1, \ldots, m$. As the purpose is to obtain a final co-clustering having the properties of $\widetilde{\mathbf{Q}} = (\widetilde{\mathbf{q}}_{ij})$, we propose to rely on a consensus structured affinity matrix $\widetilde{\mathbf{Q}}$ such that each $\widetilde{\mathbf{M}}_l$ can be modeled as $\widetilde{\mathbf{M}}_l = \widetilde{\mathbf{Q}} + E_l,\ l = 1, \ldots, m$. This leads us to the following optimization problem.

$$\min_{\widetilde{\mathbf{Q}}} \sum_{l=1}^{m} D(\widetilde{\mathbf{M}}_l, \widetilde{\mathbf{Q}}) \tag{3}$$

where D is a cost function that allows us to quantify the quality of the approximation of $\widetilde{\mathbf{M}}_l$ by $\widetilde{\mathbf{Q}}$; D can be, for instance, the Frobenius norm. This measure minimizes the disagreements between each basic co-clustering $\widetilde{\mathbf{M}}_l$ and the consensus co-clustering $\widetilde{\mathbf{Q}}$; it can be solved by $\min_{\widetilde{\mathbf{Q}}} \sum_{l=1}^{m} ||\widetilde{\mathbf{M}}_l - \widetilde{\mathbf{Q}}||_F^2$. On the other hand, it is easy to show that optimal solution $\widetilde{\mathbf{Q}}^*$ is the consensus (average) affinity matrix $\overline{\mathbf{M}} = \frac{1}{m}\sum_{l=1}^{m} \widetilde{\mathbf{M}}_l$. Hence, given $\overline{\mathbf{M}}$, the objective function to optimize becomes

$$\min_{\widetilde{\mathbf{Q}}} \mathcal{J}_{EBCO}(\overline{\mathbf{M}}, \widetilde{\mathbf{Q}}) \equiv \min_{\widetilde{\mathbf{Q}}} ||\overline{\mathbf{M}} - \widetilde{\mathbf{Q}}||_F^2. \tag{4}$$

Proposition 3.1. *Let $\overline{\mathbf{M}}$ and $\widetilde{\mathbf{Q}}$ be $n \times d$ matrices, we have,*

$$\min_{\widetilde{\mathbf{Q}}} ||\overline{\mathbf{M}} - \widetilde{\mathbf{Q}}||_F^2 \quad \equiv \quad \max_{\widetilde{\mathbf{Q}}} Tr(\overline{\mathbf{M}}\widetilde{\mathbf{Q}}^\top) \equiv \max_{\widetilde{\mathbf{Z}}, \widetilde{\mathbf{W}}} Tr(\widetilde{\mathbf{Z}}^\top\overline{\mathbf{M}}\widetilde{\mathbf{W}}) \tag{5}$$

Proof. Let us expand the left term in (5),

$$||\overline{\mathbf{M}} - \widetilde{\mathbf{Q}}||_F^2 = ||\overline{\mathbf{M}}||_F^2 + ||\widetilde{\mathbf{Q}}||_F^2 - 2Tr(\overline{\mathbf{M}}\widetilde{\mathbf{Q}}^\top) \tag{6}$$

First of all, $||\overline{\mathbf{M}}||_F^2$ is known, and does not depend on $\widetilde{\mathbf{Q}}$. On the other hand, as regards $||\widetilde{\mathbf{Q}}||_F^2$, it is easy to show that

$$||\widetilde{\mathbf{Q}}||_F^2 = Tr(\widetilde{\mathbf{Q}}\widetilde{\mathbf{Q}}^\top) = Tr(\widetilde{\mathbf{Z}}^\top\widetilde{\mathbf{Z}}\widetilde{\mathbf{W}}^\top\widetilde{\mathbf{W}}) = Tr(\mathbf{I}) = g$$

where g is the number of co-clusters that is assumed to be known (its assessment will be discussed later). Hence, due to (6) and the property $Tr(AB) = Tr(BA)$ provided that the product is possible, we have,

$$\begin{aligned} \min_{\widetilde{\mathbf{Q}}} ||\overline{\mathbf{M}} - \widetilde{\mathbf{Q}}||_F^2 \equiv \max_{\widetilde{\mathbf{Q}}} Tr(\overline{\mathbf{M}}\widetilde{\mathbf{Q}}^\top) \quad &\equiv \quad \max_{\widetilde{\mathbf{Z}}, \widetilde{\mathbf{W}}} Tr(\overline{\mathbf{M}}\widetilde{\mathbf{W}}\widetilde{\mathbf{Z}}^\top) \\ &\equiv \quad \max_{\widetilde{\mathbf{Z}}, \widetilde{\mathbf{W}}} Tr(\widetilde{\mathbf{Z}}^\top\overline{\mathbf{M}}\widetilde{\mathbf{W}}). \end{aligned}$$

□

[2]By co-partition, we mean a set of mutually exclusive and collectively exhaustive co-clusters such that each document and term are in one and only one co-cluster. Note that a co-partition is a block seriation relation.

In the following, we will see the interest of this formulation of the objective function both in terms of optimization and connections with other algorithms.

3.3 Optimization and algorithm

Before solving (5), note that using both properties $Tr(AB) = Tr(BA)$ and $Tr(A^\top) = Tr(A)$, if A is a square matrix, it easy to show that,

$$(a) \quad Tr(\widetilde{\mathbf{Z}}^\top \overline{\mathbf{M}}\widetilde{\mathbf{W}}) = Tr(\mathbf{Z}^\top \overline{\mathbf{M}}\widetilde{\mathbf{W}}\mathbf{D}_{\mathbf{z}}^{-0.5}) = \langle \overline{\mathbf{M}}\widetilde{\mathbf{W}}, \mathbf{Z}\rangle_{\mathbf{D}_{\mathbf{z}}^{-0.5}} \tag{7}$$

$$(b) \quad Tr(\widetilde{\mathbf{Z}}^\top \overline{\mathbf{M}}\widetilde{\mathbf{W}}) = Tr(\mathbf{W}^\top \overline{\mathbf{M}}^\top \widetilde{\mathbf{Z}}\mathbf{D}_{\mathbf{w}}^{-0.5}) = \langle \overline{\mathbf{M}}^\top \widetilde{\mathbf{Z}}, \mathbf{W}\rangle_{\mathbf{D}_{\mathbf{w}}^{-0.5}}. \tag{8}$$

The two terms $\langle \overline{\mathbf{M}}\widetilde{\mathbf{W}}, \mathbf{Z}\rangle_{\mathbf{D}_{\mathbf{z}}^{-0.5}}$ and $\langle \overline{\mathbf{M}}^\top \widetilde{\mathbf{Z}}, \mathbf{W}\rangle_{\mathbf{D}_{\mathbf{w}}^{-0.5}}$ in (7) and (8) suggest to replace the maximization of $Tr(\widetilde{\mathbf{Z}}^\top \overline{\mathbf{M}}\widetilde{\mathbf{W}})$ by the alternated maximization of

$$\max_{\mathbf{Z}} \langle \overline{\mathbf{M}}\widetilde{\mathbf{W}}, \mathbf{Z}\rangle_{\mathbf{D}_{\mathbf{z}}^{-0.5}} \text{ and } \max_{\mathbf{W}} \langle \overline{\mathbf{M}}^\top \widetilde{\mathbf{Z}}, \mathbf{W}\rangle_{\mathbf{D}_{\mathbf{w}}^{-0.5}}.$$

Given an initial guess of $\mathbf{W}$ and $\mathbf{Z}$, we iteratively update the model parameters,

Update of Z: when $\mathbf{W}$ is fixed, $\mathbf{Z}$ can be obtained by maximizing $\langle \overline{\mathbf{M}}\widetilde{\mathbf{W}}, \mathbf{Z}\rangle_{\mathbf{D}_{\mathbf{z}}^{-0.5}} = \sum_{i,k} z_{ik} \frac{1}{\sqrt{z_{.k}}} \widetilde{\mathbf{w}}_k^\top \overline{\mathbf{m}}_i$. Then the update of $\mathbf{Z}$ is given by $\forall i, z_{ik} = \text{argmax}_{k'} \frac{1}{\sqrt{z_{.k'}}} \widetilde{\mathbf{w}}_{k'}^\top \overline{\mathbf{m}}_i \in \{0,1\}$, this leads to

$$\mathbf{Z} = Binmax^3(\overline{\mathbf{M}}\widetilde{\mathbf{W}}\mathbf{D}_{\mathbf{z}}^{-0.5}). \tag{9}$$

Update of W: when $\mathbf{Z}$ is fixed, $\mathbf{W}$ can be obtained by maximizing $\langle \overline{\mathbf{M}}^\top \widetilde{\mathbf{Z}}, \mathbf{W}\rangle_{\mathbf{D}_{\mathbf{w}}^{-0.5}} = \sum_{j,k} w_{jk} \frac{1}{\sqrt{w_{.k}}} \widetilde{\mathbf{z}}_k^\top \overline{\mathbf{m}}^j$. Then the update of $\mathbf{W}$ is given by $\forall j, w_{jk} = \text{argmax}_{k'} \frac{1}{\sqrt{w_{.k'}}} \widetilde{\mathbf{z}}_{k'}^\top \overline{\mathbf{m}}^j \in \{0,1\}$. This leads to

$$\mathbf{W} = Binmax(\overline{\mathbf{M}}^\top \widetilde{\mathbf{Z}}\mathbf{D}_{\mathbf{w}}^{-0.5}). \tag{10}$$

Note that the update rules show the mutual interaction between the set of documents and the set of terms. In the sequel, we give the details of the alternating procedure of EBCO in Algorithm 1.

Algorithm 1 Ensemble Block Co-Clustering (EBCO).

Input: $\mathcal{M} = \{(\mathbf{Z}_l, \mathbf{W}_l); l = 1, \ldots, m\}$ a collection of co-partitions, g number of co-clusters,
Output: co-partition $(\mathbf{Z}, \mathbf{W})$
Initialization:
a) Compute $\overline{\mathbf{M}}$
b) Random initialization of $\mathbf{W}$ and $\mathbf{Z}$[4].
repeat
1. Assignment of documents (9)
• $\mathbf{Z} \leftarrow Binmax(\overline{\mathbf{M}}\widetilde{\mathbf{W}}\mathbf{D}_{\mathbf{z}}^{-0.5})$
2. Assignment of terms (10)
• $\mathbf{W} \leftarrow Binmax(\overline{\mathbf{M}}^\top \widetilde{\mathbf{Z}}\mathbf{D}_{\mathbf{w}}^{-0.5})$
until convergence of $J_{EBCO}(\overline{\mathbf{M}}, \widetilde{\mathbf{Q}}) = ||\overline{\mathbf{M}} - \widetilde{\mathbf{Q}}||_F^2$ (4)

Note that $\overline{\mathbf{M}}\widetilde{\mathbf{W}}\mathbf{D}_{\mathbf{z}}^{-0.5}$ and $\overline{\mathbf{M}}^\top \widetilde{\mathbf{Z}}\mathbf{D}_{\mathbf{w}}^{-0.5}$ are the projections of documents and words in the low-dimensional space. Furthermore, we observe the *conscience mechanism* principle [6] [5] thanks to the role played by the diagonal matrices $\mathbf{D}_{\mathbf{z}}^{-0.5}$ and $\mathbf{D}_{\mathbf{w}}^{-0.5}$. Besides, EBCO is computationally efficient and its complexity can be shown to be $O(n \cdot it \cdot (2g))$ where it is the number of iterations, which is small (about few dozens).

[3] Let $\mathbf{A} = (a_{ik}) \in \{0,1\}^{n \times g}$ with $\forall i, \sum_k a_{ik} = 1$ and $\mathbf{B} = (b_{ik}) \in \mathbb{R}^{n \times g}$, then $\mathbf{A} \leftarrow Binmax(\mathbf{B})$ means $\forall i, a_{ik} = \text{argmax}_{k'} b_{ik'}, k' = 1, \ldots, g$.
[4] Initial $\mathbf{W}$ and $\mathbf{Z}$ can be obtained, for instance, by using spherical k-means.

4 RELATIONS AMONG DIFFERENT CONSENSUS AND CO-CLUSTERING FUNCTIONS

We detail in this section the connections with various criteria. In particular, we highlight the equivalence relation between our proposal and other state-of-the-art consensus and co-clustering methods devoted to text data – namely one side ensemble clustering, spectral co-clustering, NMTF and double spherical k-means –. The mathematical details of derivation on the connections are presented in the rest of this section.

4.1 Connection to ensemble clustering

First of all, we show that the proposed framework for ensemble co-clustering is a natural generalization of the well studied one side ensemble clustering. In fact, the proposed optimization problem in Eq. (4) generalizes the ensemble clustering objective based on the co-association matrix. Indeed, the consensus co-clustering modeled by a scaled Block seriation relation $\widetilde{\mathbf{Q}} = \widetilde{\mathbf{Z}}\widetilde{\mathbf{W}}^\top$ which is defined on $I \times J$, generalizes the scaled equivalence relation $\widetilde{\mathbf{Y}} = \widetilde{\mathbf{Z}}\widetilde{\mathbf{Z}}^\top$, which is defined on $I \times I$. Let us consider the objective function of EBCO,

$$\min_{\widetilde{\mathbf{Q}}} \sum_{l=1}^{m} ||\widetilde{\mathbf{M}}_l - \widetilde{\mathbf{Q}}||_F^2 \quad \equiv \quad \min_{\widetilde{\mathbf{Q}}} ||\overline{\mathbf{M}} - \widetilde{\mathbf{Q}}||_F^2.$$

When $I \equiv J$, $\widetilde{\mathbf{Q}}$ becomes $\widetilde{\mathbf{Y}}$ and $\overline{\mathbf{M}}$ takes the following form $\bar{\mathbf{S}} = \frac{1}{m}\sum_{l=1}^{m} \widetilde{\mathbf{Z}}_l \widetilde{\mathbf{Z}}_l^\top$. Thereby, the objective function optimized by EBCO is reduced to the objective of one side ensemble clustering as follows,

$$\min_{\widetilde{\mathbf{Y}}} \sum_{l=1}^{m} ||\widetilde{\mathbf{S}}_l - \widetilde{\mathbf{Y}}||_F^2 \equiv \min_{\widetilde{\mathbf{Y}}} ||\bar{\mathbf{S}} - \widetilde{\mathbf{Y}}||_F^2 \equiv \max_{\widetilde{\mathbf{Z}}^\top\widetilde{\mathbf{Z}}=\mathbf{I}} Tr(\widetilde{\mathbf{Z}}^\top \bar{\mathbf{S}}\widetilde{\mathbf{Z}}).$$

Note that the combined co-affinity matrix $\overline{\mathbf{M}}$ can be viewed as an extension of $\bar{\mathbf{S}}$ which is central in the ensemble clustering context. Furthermore, all the connections which will be discussed below remain valid in the context of ensemble clustering.

4.2 Connection to spectral co-clustering

In the optimization problem of Eq. (4), we relax the non-negativity constraint on both $\widetilde{\mathbf{Z}}$ and $\widetilde{\mathbf{W}}$. Therefore, we have,

$$\min_{\widetilde{\mathbf{Q}}} \mathcal{J}_{EBCO}(\widetilde{\mathbf{Q}}) \quad \equiv \quad \max_{\widetilde{\mathbf{Z}}^\top\widetilde{\mathbf{Z}}=\mathbf{I}, \widetilde{\mathbf{W}}^\top\widetilde{\mathbf{W}}=\mathbf{I}} Tr(\widetilde{\mathbf{Z}}^\top \overline{\mathbf{M}}\widetilde{\mathbf{W}})$$

where $\widetilde{\mathbf{Z}} = \mathbf{Z}\mathbf{D}_{\mathbf{z}}^{-0.5}$ and $\widetilde{\mathbf{W}} = \mathbf{W}\mathbf{D}_{\mathbf{w}}^{-0.5}$. It is easy to verify that $\widetilde{\mathbf{Z}}$ and $\widetilde{\mathbf{W}}$ satisfy the orthogonality constraint, i.e. $\widetilde{\mathbf{Z}}^\top\widetilde{\mathbf{Z}} = \mathbf{I}$ and $\widetilde{\mathbf{W}}^\top\widetilde{\mathbf{W}} = \mathbf{I}$. This optimization problem can be performed by Lagrange multipliers into eigenvalue problem. Then, given $svd(\overline{\mathbf{M}}) = \widetilde{\mathbf{Z}}\Sigma\widetilde{\mathbf{W}}^\top$, the discrete co-clustering is obtained by performing k-means on

[5] When the clusters are by nature very unbalanced, the conscience mechanism has a regularizing effect that makes it possible to escape poor locally optimal solutions where some clusters are very big/small or even empty.

the concatenated data $\left[\widetilde{\mathbf{Z}}\ \widetilde{\mathbf{W}}\right]^\top$. This is equivalent to the spectral co-clustering method proposed in [7].

4.3 Connection to NMTF

In a similar way, we can establish a connection with Non-negative Matrix Tri-Factorization (NMTF) [23, 36, 37] that consists in approximating $\mathbf{X}$ by $\mathbf{ZDW}^\top$. Let us consider the weighting diagonal matrix $\mathbf{D}_{(g\times g)} = (\mathbf{d}_{kk})$ defined by $\mathbf{D} = \mathbf{D}_\mathbf{z}^{-0.5}\mathbf{D}_\mathbf{w}^{-0.5}$; $\mathbf{d}_{kk} = \frac{1}{\sqrt{z_{.k}\, w_{.k}}}$ depends on a geometric mean of documents and words cluster sizes in co-cluster kk, then we have

$$\begin{aligned}\min_{\mathbf{Z},\mathbf{W}} ||\mathbf{X} - \mathbf{Z}\mathbf{D}_\mathbf{z}^{-0.5}\mathbf{D}_\mathbf{w}^{-0.5}\mathbf{W}^\top||_F^2 &\equiv \min_{\mathbf{Z},\mathbf{W},\mathbf{D}=\mathbf{D}_\mathbf{z}^{-0.5}\mathbf{D}_\mathbf{w}^{-0.5}} ||\mathbf{X} - \mathbf{ZDW}^\top||_F^2 \\ &\equiv \max_{\mathbf{Z},\mathbf{W},\mathbf{D}} Tr(\mathbf{Z}^\top\mathbf{XWD}) \\ &\equiv \max_{\widetilde{\mathbf{Q}}} \mathcal{J}_{EBCO}(\mathbf{X},\widetilde{\mathbf{Q}}).\end{aligned}$$

Thereby, the criterion optimized by Fast NMTF proposed in [37] applied on $\mathbf{X}$ with an additional constraint on the centroid matrix $\mathbf{D}$ is equivalent to EBCO applied on $\mathbf{X}$.

4.4 Connection to double weighted spherical k-means

Note that as $\overline{\mathbf{M}}$ can be written in vector form in two ways: $\overline{\mathbf{M}} = [\overline{\mathbf{m}}_1, \ldots, \overline{\mathbf{m}}_n]^\top$ or $\overline{\mathbf{M}} = [\overline{\mathbf{m}}^1, \ldots, \overline{\mathbf{m}}^d]$, we can derive two expressions of $Tr(\widetilde{\mathbf{Z}}^\top\overline{\mathbf{M}}\widetilde{\mathbf{W}}) = \langle\overline{\mathbf{M}}\widetilde{\mathbf{W}}, \mathbf{Z}\rangle_{\mathbf{D}_\mathbf{z}^{-0.5}} = \langle\overline{\mathbf{M}}^\top\widetilde{\mathbf{Z}}, \mathbf{W}\rangle_{\mathbf{D}_\mathbf{w}^{-0.5}}$. Indeed from Eq. (7) and Eq. (8) respectively we have,

$$\langle\overline{\mathbf{M}}\widetilde{\mathbf{W}}, \mathbf{Z}\rangle_{\mathbf{D}_\mathbf{z}^{-0.5}} = \sum_{i,k} z_{ik}\frac{1}{\sqrt{z_{.k}}}\widetilde{\mathbf{w}}_k^\top\overline{\mathbf{m}}_i = \sum_{i,k}\frac{1}{\sqrt{z_{.k}}}z_{ik}cos(\widetilde{\mathbf{w}}_k, \overline{\mathbf{m}}_i),$$

$$\langle\overline{\mathbf{M}}^\top\widetilde{\mathbf{Z}}, \mathbf{W}\rangle_{\mathbf{D}_\mathbf{w}^{-0.5}} = \sum_{j,k} w_{jk}\frac{1}{\sqrt{w_{.k}}}\widetilde{\mathbf{z}}_k^\top\overline{\mathbf{m}}^j = \sum_{j,k}\frac{1}{\sqrt{w_{.k}}}w_{jk}cos(\widetilde{\mathbf{z}}_k, \overline{\mathbf{m}}^j).$$

From both formulations, we can observe that EBCO looks like a double weighted spherical k-means with a conscience mechanism devoted to document clustering and referred to as DCC [26]. Both criteria are weighted by $1/\sqrt{z_{.k}}$ and $1/\sqrt{w_{.k}}$ whose role here is to discourage larger clusters to absorb new rows or columns and therefore, to avoid empty clusters. However, unlike DCC, with EBCO the norms of $\widetilde{\mathbf{z}}_k$, $\widetilde{\mathbf{w}}_k$, $\overline{\mathbf{m}}^j$ and $\overline{\mathbf{m}}_i$ are not necessarily equal to 1.

5 UNSUPERVISED BASIC CO-CLUSTERINGS SELECTION AND ASSESSMENT OF THE NUMBER OF CO-CLUSTERS

A challenging problem in co-clustering is the inference of the number of co-clusters which is often assumed to be known by the user. In this section, we present a method to determine this parameter based on the work of [17], where the authors rely on modularity criterion to deal with co-clustering of categorical data.

5.1 EBCO objective versus Modularity criterion

Given the affinity matrix $\overline{\mathbf{M}}$ defined on $I \times J$, to tackle the co-clustering we consider the following generalized modularity measure $\mathcal{J}_{Mod}$ defined in [17],

$$\mathcal{J}_{Mod}(\mathbf{Q}) = \frac{1}{2|E|}\sum_{i=1}^{n}\sum_{j=1}^{n}(\overline{m}_{ij} - \frac{\overline{m}_{i.}\overline{m}_{.j}}{2|E|})\mathbf{q}_{ij}. \tag{11}$$

where $2|E| = \sum_{i,j}\overline{m}_{ij} = \overline{m}_{..}$ is the total weight of edges and $\overline{m}_{i.} = \sum_j \overline{m}_{ij}$ - the degree of i and $\overline{m}_{.j} = \sum_i \overline{m}_{ij}$- the degree of j. This Modularity measure takes the following matrix trace form,

$$\mathcal{J}_{Mod}(\mathbf{Q}) = \frac{1}{2|E|}Tr[(\overline{\mathbf{M}} - \delta)^\top\mathbf{Q}] \text{ where } \delta_{ij} = \frac{\overline{m}_{i.}\overline{m}_{.j}}{\overline{m}_{..}}. \tag{12}$$

As the objective function in Eq. (12) is linear with respect to $\mathbf{Q}$ and as the constraints that Z must respect linear equations, we can theoretically solve the problem using an integer linear programming solver. However, this problem is *NP* hard; as result we use heuristics in practice for dealing with large datasets.

In Eq. (12), if we set $\delta = 0$ and consider the scaled Block seriation $\widetilde{\mathbf{Q}}$ instead of the binary Block seriation $\mathbf{Q}$, $\mathcal{J}_{Mod}$ is equivalent to $\mathcal{J}_{EBCO}$. Then, when $\delta = 0$, EBCO can be viewed as a relaxation of the modularity criterion which considers the size of row clusters and column clusters, and leads to a tractable optimization problem.

5.2 Assessing of the number of co-clusters

Since modularity objective is not a trivial criterion, we co-cluster each data set into different number of co-clusters varying from 2 to K. For each fixed number of co-clusters, the co-clustering modularity is computed and the optimal number of co-clusters is considered to correlate well with the maximum modularity value. Unlike to known spectral clustering methods, in [2] authors show that the modularity measure allows natural co-clusters identification, i.e. the maximum value of modularity correlates well with the optimal number of co-clusters. EBCO enables a modularity-based evaluation of the number of co-clusters at the step denoted as *Option 2* in Figure 1.

5.3 Unsupervised selection of relevant basic co-clusterings

Given a collection of basic co-clusterings generated on the same dataset, the co-clustering modularity is computed for each basic co-clustering and the selection of relevant co-clusterings in the final collection is considered based on the modularity value. We consider only basic co-clusterings with high modularity value (modularity value greater than a fixed threshold), i.e the relevance of a co-clustering correlate well with the maximum modularity value. EBCO enables a modularity-based selection of the basic co-clusterings at the step denoted as *Option 1* in Figure 1.

6 EXPERIMENTS

We performed extensive experiments on a wide range of *real-world* text datasets. Our results demonstrate the high performance of EBCO . In particular, we compare our approach with four effective diagonal and non-diagonal co-clustering algorithms that are very competitive in the field of text co-clustering, namely DCC [26], CoClustMod [1, 2], CoClustSpecMod [17] and CROINFO [11, 12].

We also compare EBCO to CoCE [39], a recent and competitive co-clustering ensemble algorithm. For CROINFO[6], CoClustMod and CoClustSpecMod, we used the implementation and default parameters of the CoClust Python package [25]. For CoCE, we used the Matlab source code proposed by the authors[7]. EBCO and DCC were implemented in Python. Note that in [1, 2], through extensive experiments on text datasets, the authors showed that CoclusMod outperforms several other notable co-clustering methods that we do not retain in our comparisons. Similarly, as the competitiveness of CoCE was extensively demonstrated in [39], we do not include the approaches outperformed by CoCE in our experiments.

6.1 Benchmark datasets

For our evaluations, we consider 7 benchmark document-term datasets that are popular for the document clustering task, namely SPORTS, TR45, PUBMED10, LA12, CLASSIC4, CSTR and CLASSIC3 (Table 1). Each document-term data matrix $\mathbf{X}$ can be viewed as a contingency table (or a two-way frequency table) where x_{ij} indicates the number of occurrences of word j in document i. Together, these datasets embed several challenging situations such as different degrees of cluster balance, diverse cluster sizes and various degrees of cluster overlap. These datasets cover a wide range of imbalance strength as can be seen from their *Balance* coefficient (Table 1), which is the ratio of the minimum cluster size to the maximum cluster size. As is frequently the case in document-term co-clustering, the labels of benchmark datasets are only known for documents and not the words. Yet, as the words partition is inherently associated with the document partition, we expect that the quality of the document clustering is informative about the quality of the word clustering.

Table 1: Description of Datasets

Datasets	Characteristics				
	#documents	#words	g	Sparsity (%)	Balance
SPORTS	8580	14870	7	99.14	0.036
TR45	690	8261	10	96.60	0.088
PUBMED10	15565	22437	10	99.72	0.093
LA12	6279	31472	6	99.52	0.281
CLASSIC4	7094	5896	4	99.41	0.323
CSTR	475	1000	4	96.60	0.399
CLASSIC3	3891	4303	3	98.95	0.710

6.2 Experimental settings

We normalized each document-term matrix using the TF-IDF weighting scheme (*term-frequency* times *inverse document frequency*) as implemented in the scikit-learn Python package. The EBCO results are averaged over 10 different runs. Each run involves the generation of an ensemble matrix $\overline{\mathbf{M}}$ based on several basic co-clusterings with a growing number of co-clusters (Fig. 1, steps (a) & (b)) and a final co-clustering on the ensemble matrix $\overline{\mathbf{M}}$ with 10 different initializations and 100 iterations (Fig. 1, step (c)). For each run, the final co-clustering that corresponds to the best algorithm criterion is automatically retained. Furthermore, to avoid poor local solution that could be induced by early hard word assignments in the iteration, we perform stochastic column assignments during the first 70 iterations, as described in [27].

[6] CROINFO and ITCC [8] are equivalent, they optimise the same objective function. Further, CROINFO is a new formulation of ITCC in the spirit of clustering algorithms, both are linked to a restricted Poisson Latent Block Model [12]

[7] http://mlda.swu.edu.cn/codes.php?name=CoCE.

6.3 Evaluation metrics

To evaluate the performance of EBCO and the competitive text co-clustering methods (DCC, CoClustMod, CoClustSpecMod and CROINFO), we compare the document clustering of their bi-partitions with the original document labels. The comparisons are made by computing the *Adjusted Rand Index* (ARI) [16] and the *Normalized Mutual Information* (NMI) [28], which are two widely used measures that assess the similarity between the estimated clusters and the true clustering. In particular, NMI evaluates how the estimated clustering is informative about the known clustering, and ARI quantifies the agreement between the estimated clustering and the true labels. NMI, unlike ARI, is less sensitive to cluster splitting or merging. For both metrics, we used the implementation provided by the scikit-learn Python package.

In Section 6.4, we first evaluate the EBCO performance on a *complete collection* of $s = 24$ basic co-clusterings with a growing number of co-clusters from 2 to 25 and using the original document cluster number g as final number of co-clusters. We compare these evaluations to the results obtained with several recent and competitive text co-clustering methods, namely DCC, CoClustMod, CoClustSpecMod and CROINFO. Then, in Section 6.5, we evaluate the EBCO performance on a *partial collection* where the basic co-clusterings are automatically selected based on modularity ($s < 24$; Fig. 1, Option (1)). Finally, we evaluate in Section 6.6 the fully unsupervised version of EBCO, when the final number of co-clusters is also automatically learned based on modularity (Fig. 1, Option (2)).

6.4 *Complete* co-clusterings collection ensemble

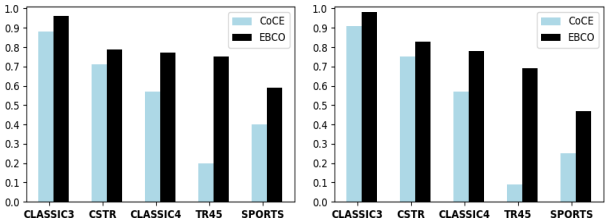
As detailed in Section 6.1, only the document labels are known and are used to evaluate the different approaches in this section. We first obtained a collection of basic co-clusterings, $\{(\mathbf{Z}_l, \mathbf{W}_l)\}_{l\in[2...25]}$. Then, these co-clusterings are merged into $\overline{\mathbf{M}}$, before repeating the final co-clustering alternating steps until the convergence of the objective function $\mathcal{J}_{EBCO}$ (Algorithm 1). While the number of co-clusters for each basic co-clustering varies from 2 to 25, the final number of co-clusters is set to the known number g for each benchmark document-text dataset. The NMI and ARI results are average over 10 trials. Table 2 summarizes the comparative results of EBCO for the 7 datasets.

As can be seen from Table 2, our co-clustering ensemble approach outperforms the other methods on all datasets with a significant margin. Interestingly, EBCO outperforms both the ARI and NMI of the other approaches, with an average ARI and NMI increase of 0.128 and 0.098 respectively. These improvements, in particular for ARI, shows the good capacity for EBCO to handle imbalanced clusters.

In figure 2 we note the high performance of EBCO as compared to CoCE in terms of clustering. Furthermore, it should also be remembered that unlike EBCO having a linear complexity, the computational complexity of CoCE is dominated by the execution of

Table 2: Mean±sd clustering NMI and ARI. Bold values indicate the best result over all methods.

Datasets		DCC	CoClustMod	CoClustSpecMod	CROINFO	EBCO
SPORTS	NMI	0.57 ± 0.01	0.53 ± 0.04	0.45 ± 0.00	0.57 ± 0.03	**0.59 ± 0.01**
	ARI	0.39 ± 0.01	0.45 ± 0.06	0.30 ± 0.00	0.45 ± 0.04	**0.47 ± 0.06**
TR45	NMI	0.69 ± 0.02	0.49 ± 0.03	0.71 ± 0.02	0.54 ± 0.03	**0.75 ± 0.01**
	ARI	0.56 ± 0.04	0.42 ± 0.02	0.66 ± 0.05	0.44 ± 0.06	**0.69 ± 0.02**
PUBMED10	NMI	0.56 ± 0.01	0.49 ± 0.02	0.36 ± 0.00	0.56 ± 0.03	**0.63 ± 0.01**
	ARI	0.44 ± 0.01	0.44 ± 0.04	0.18 ± 0.00	0.46 ± 0.04	**0.52 ± 0.02**
LA12	NMI	0.52 ± 0.02	0.42 ± 0.02	0.35 ± 0.00	0.45 ± 0.04	**0.59 ± 0.01**
	ARI	0.45 ± 0.03	0.41 ± 0.03	0.17 ± 0.00	0.38 ± 0.05	**0.57 ± 0.02**
CLASSIC4	NMI	0.72 ± 0.00	0.67 ± 0.03	0.41 ± 0.00	0.66 ± 0.06	**0.77 ± 0.01**
	ARI	0.71 ± 0.01	0.61 ± 0.05	0.22 ± 0.00	0.58 ± 0.11	**0.78 ± 0.01**
CSTR	NMI	0.68 ± 0.01	0.66 ± 0.05	0.76 ± 0.00	0.69 ± 0.02	**0.79 ± 0.00**
	ARI	0.61 ± 0.05	0.70 ± 0.05	0.80 ± 0.00	0.70 ± 0.07	**0.83 ± 0.01**
CLASSIC3	NMI	0.94 ± 0.00	0.94 ± 0.00	0.91 ± 0.00	0.94 ± 0.00	**0.96 ± 0.00**
	ARI	0.97 ± 0.00	0.97 ± 0.00	0.94 ± 0.00	0.96 ± 0.00	**0.98 ± 0.00**



(a) NMI (b) ARI

Figure 2: Comparative evaluations for CoCE and EBCO on text datasets.

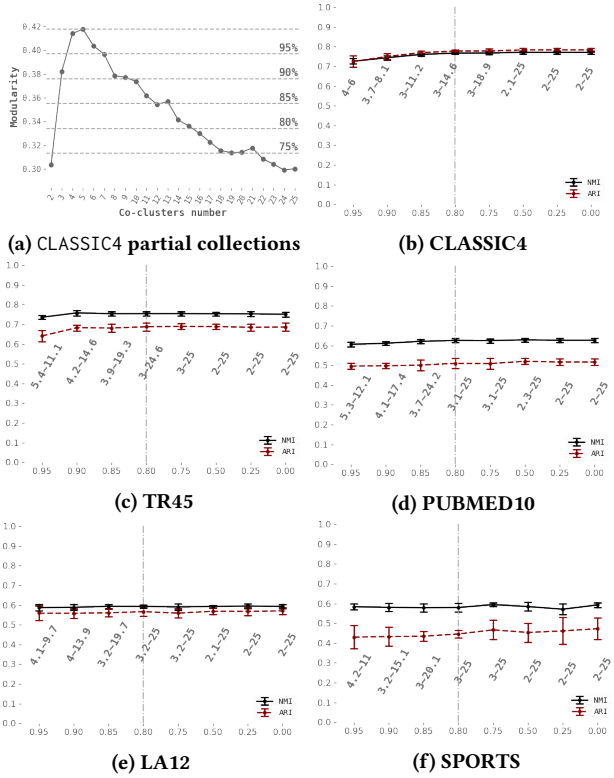
SVD; which is $O(eNnd)$, where $e = \lceil \log_2 g \rceil$ and N is the number of 50 Lanczos iteration steps as mentioned in [39]. Such complexity makes the use of CoCE quite prohibitive for high dimensional text data.

6.5 *Partial* co-clusterings collection ensemble

Using the complete basic co-clusterings collection already provides a significant improvement of EBCO as compare to alternative text co-clustering approaches or ensemble co-clustering methods (Section 6.4). However, not all the basic co-clusterings are necessary to build the joint matrix $\overline{\mathbf{M}}$. In fact, based on a modularity criteria, one can identify the *best* basic co-clustering from the whole collection and its closest co-clusterings that could bring valuable information. Specifically, taking the highest modularity of the collection, we can select a *partial collection* of models that reach a modularity equal or greater than 95%, 90%, 85%, 80%, 75%, 50% or 25% the maximum modularity. The Figure 3a exemplifies this selection for CLASSIC4 (horizontal dashed lines).

Figure 3b to Figure 3f illustrate the evolution of NMI and ARI when increasing the partial collection up to the complete collection for several benchmark datasets. Our experiments indicate that a partial collection of models with a modularity value of at least 80% of the maximum modularity would be sufficient to reach the maximum NMI and ARI. Therefore, we would generally advice this percentage for other studies.

Interestingly, one can see from Figure 3a that even if the basic co-clusterings infer a number of co-clusters of 5 instead of 4 for the CLASSIC4 dataset, the fully unsupervised version of EBCO ultimately



(a) CLASSIC4 partial collections (b) CLASSIC4

(c) TR45 (d) PUBMED10

(e) LA12 (f) SPORTS

Figure 3: (a), Partial collection selection for CLASSIC4 using modularity. Dashed lines indicate the maximum modularity percentage required. (b) to (e), Mean±sd clustering NMI and ARI by percentage of maximum modularity. Gray annotations give the range of co-cluster number.

infer the expected number of co-clusters (see Section 6.6, Table 3). This reinforce the idea that beyond the model that maximizes the modularity, closest models should also be considered in an ensemble approach as they can bring relevant information.

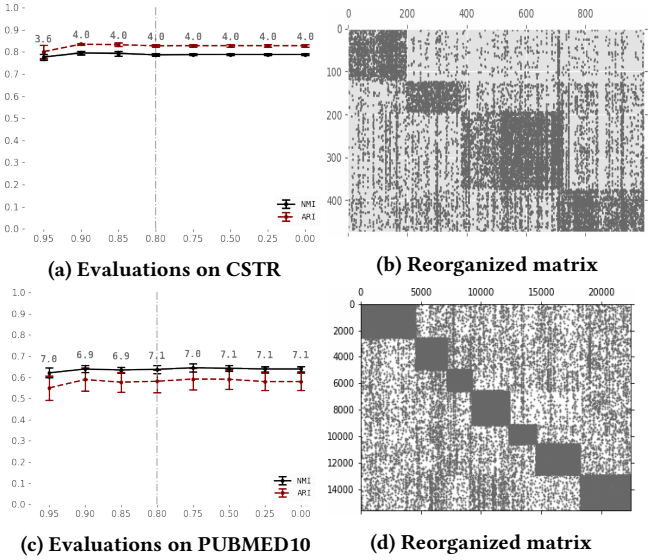
6.6 Fully unsupervised ensemble co-clustering

We evaluate in this section the ability of EBCO to run as a fully unsupervised approach (Fig. 1, EBCO with Option 1 *and* Option 2). Specifically, EBCO can automatically infer the best number of co-clusters based on modularity (Fig. 1, Option (2)). Table 3 gives the NMI and ARI results for EBCO and the inferred number of co-clusters $g^\star$ for the 7 benchmark datasets. As empirically suggested by our experiments, we have set the partial collection with models having at least 80% of the maximum modularity value.

As can be seen in Table 3, EBCO infers a number of co-clusters $g^\star$ that is equal or closed to the expected number g for almost all benchmark datasets. For one dataset, PUBMED10, the number of co-clusters is more strongly underestimated. We hypothesis that for this dataset the inferred number of co-clusters $g^\star$ better reflects the *actual* co-cluster number as several document topics are inherently intertwined. We specifically discuss this point in details in Section 6.7.

Table 3: Mean±sd clustering NMI and ARI. Basic co-clusterings modularity is at least 80% of the maximum collection modularity.

Datasets		$\text{EBCO}_{80\%}$	$g^\star$	Expected g
CLASSIC3	NMI	0.95 ± 0.00	3.0	3
	ARI	0.97 ± 0.00		
CSTR	NMI	0.79 ± 0.00	4.0	4
	ARI	0.83 ± 0.01		
CLASSIC4	NMI	0.76 ± 0.02	4.1	4
	ARI	0.76 ± 0.06		
LA12	NMI	0.61 ± 0.02	5.7	6
	ARI	0.59 ± 0.03		
SPORTS	NMI	0.59 ± 0.02	6.0	7
	ARI	0.51 ± 0.06		
TR45	NMI	0.76 ± 0.02	8.5	10
	ARI	0.70 ± 0.04		
PUBMED10	NMI	0.64 ± 0.02	7.1	10
	ARI	0.58 ± 0.03		



(a) Evaluations on CSTR

(b) Reorganized matrix

(c) Evaluations on PUBMED10

(d) Reorganized matrix

Figure 4: (a) & (c). Mean±sd clustering NMI and ARI for several partial basic co-clustering collections. Gray datapoint annotations give the inferred co-clusters number. (b) & (d). EBCO reorganized documents × terms matrices.

In Figures 4a & 4c, we give for CSTR and PUBMED10 the inferred number of co-clusters (gray datapoint annotation) while increasing the size of the partial collection. We also provide the evolution of the NMI and ARI. For both datasets, the EBCO co-clustering provides a clear partitioning of the input document-term matrices, as can be seen from Figures 4b & 4d.

6.7 Evaluating topic and word clusterings

In the following, we first analyse the co-clusters top terms for PUBMED10 [4]. We then study the distribution of document topics in the co-clusters and make the link with their top terms. In particular, we study the evolution of the topics distribution when the number of co-clusters varies from 10 (the known document labels) to 7 (the EBCO estimation). PUBMED10 is based on approximately 15,000 biomedical abstracts downloaded from Medline database that cover 10 diseases and that were published between 2000 and 2008. Each document is originally labeled with the corresponding disease (Table in Figure 5). Our results suggest that EBCO infers a number of co-clusters ($g^\star = 7$) that reflects *actual* biomedical relationships between diseases (Figure 5, bottom line).

6.7.1 EBCO *co-clusters top terms.* Identifying the most representative terms of each co-clustering that indicate the main topics is a challenging task. It is usual to provide a simple ordering of the co-cluster words in decreasing order of their frequency and select the most frequent words to characterize the co-cluster topics. Yet, this option can favor the appearance of none informative terms among the top terms, such as adverbs or pronouns.

We propose an improvement over the frequency ordering method that is based on the Normalized Pointwise Mutual Information (NPMI) to reorder the most frequent co-cluster words. The NPMI ranges between −1 and +1, and is formally defined as $\text{NPMI}(w_i, w_j) = \text{PMI}(\text{w}_\text{i}, \text{w}_\text{j})/\log(p(w_i, w_j))$, where the PMI between words w_i and w_j is defined as $\log\left(p(w_i, w_j)/p(w_i)p(w_j)\right)$. We use the NPMI through a k-nn-like (k nearest neighbors) approach to compute for each word a NPMI_i score defined a,

$$\text{NPMI}_i = \frac{1}{|\Omega_i|} \sum_{w_j \in \Omega_i} \text{NPMI}(w_i, w_j) \tag{13}$$

where Ω_i is the set of k words w_j having the highest NPMI score with w_i. Therefore, the NPMI_i quantifies the membership of a word in a cluster based on its relationships with its k most closest NPMI neighbors. The Table 4 gives the top NPMI_i words for PUBMED10 word clusters ($k = 5$ neighbors among the 30 most frequent terms) for the fully unsupervised EBCO co-clustering ($g^\star$=7). Probabilities are derived from the whole English Wikipedia, using a NPMI implementation proposed by Röder *et al.* [24]. Therefore, the NPMI_i scores are independent of the input document-term data.

As can be seen from Table 4, the top NPMI_i terms contain words that are coherent with the main topics. For instance, the **Hay Fever** 10 top terms now include *immunotherapy* – a seasonal allergy treatment – and *oesinophil* – a marker in seasonal allergic rhinitis –, while the frequency-based approach rank these terms at the 24^{th} and 25^{th} position respectively. The word *pneumonia* can be found in the **Otitis** top terms at the 2^{nd} position – instead of 19^{th} based only on frequency – which is coherent with the fact that *Streptococcus pneumoniae* is the most common microbial agent found in otitis. Finally, the **AMD** 10 top terms are enriched with *diabetic* – an AMD risk factor –, and *edema* – a symptom of macular degeneration.

6.7.2 Distribution of document topics in EBCO *co-clusters.* The Figure 5 summarizes the topics distribution for EBCO co-partitions with a number of co-clusters between $g' = 10$ (top) and $g' = 7$ (bottom). Pie charts gives the percentage of *disease-documents* associated to EBCO co-clusters. As can be seen, several co-clusters are *stable* and keep a clear predominant topic when changing the number of co-clusters, such as **AMD** (gray pie charts), **Otitis** (blue pie charts), **Migraine** (light green pie charts without **Raynaud Disease**) and **Hay Fever** (brown pie charts).

We can observe a *stable* co-cluster with two main topics, namely **Hepatitis A** and **Chickenpox** (Fig. 5, purple and orange pie charts).

Table 4: NPMI_i scores *within* top frequent word clusters for the fully unsupervised EBCO ($g^\star = 7$)

Kidney Calculi Jaundice Gout	NPMI_i	**AMD**	NPMI_i	**Otitis**	NPMI_i	**Migraine**	NPMI_i	**Migraine AMD Raynaud Disease**	NPMI_i	**Hay Fever**	NPMI_i	**Hepatitis A Chickenpox**	NPMI_i
uric	0.52	macular	0.61	otitis	0.48	placebo	0.34	mutation	0.42	allergic	0.55	varicella	0.57
kidney	0.46	degeneration	0.47	pneumonia	0.44	efficacy	0.33	gene	0.39	rhinitis	0.55	zoster	0.56
urinary	0.46	retinal	0.46	antibiotics	0.39	adverse	0.33	allele	0.39	allergy	0.54	virus	0.48
urine	0.45	edema	0.37	bacterial	0.38	treatment	0.33	genetic	0.39	asthma	0.51	vzv	0.48
oxalate	0.44	acuity	0.37	acute	0.38	dose	0.32	polymorphism	0.37	allergen	0.51	hepatitis	0.46
renal	0.44	diabetic	0.37	chronic	0.37	drug	0.31	disease	0.27	immunotherapy	0.40	infection	0.43
calcium	0.39	optic	0.33	influenza	0.32	headache	0.30	migraine	0.24	nasal	0.38	viral	0.42
serum	0.38	visual	0.30	recurrent	0.32	effect	0.30	affect	0.23	pollen	0.38	antibodies	0.36
acid	0.37	vision	0.27	effusion	0.32	pain	0.29	factor	0.23	symptom	0.34	immune	0.35
gout	0.36	eye	0.26	complication	0.31	triptan	0.25	identify	0.19	skin	0.28	prevalence	0.32
obstruction	0.32	injection	0.26	ear	0.29	treat	0.24	associate	0.19	exposure	0.25	incidence	0.28
jaundice	0.30	laser	0.23	pathogenic	0.28	severe	0.23	analysis	0.18	cell	0.25	estimate	0.27
lithotripsy	0.30	amd	0.22	resistant	0.25	medical	0.23	evidence	0.18	eosinophil	0.25	detect	0.22
patient	0.29	therapy	0.21	isolate	0.20	prevention	0.23	suggest	0.18	airway	0.23	outbreak	0.21
calculi	0.28	risk	0.21	membrane	0.20	trial	0.22	blood	0.15	seasonal	0.23	sample	0.17

Disease	#doc.
Gout	543
Chickenpox	732
Raynaud's disease	343
Jaundice	503
Hepatitis A	796
Hay Fever	1517
Kidney Calculi	1549
AMD	3283
Migraine	3703
Otitis	2596

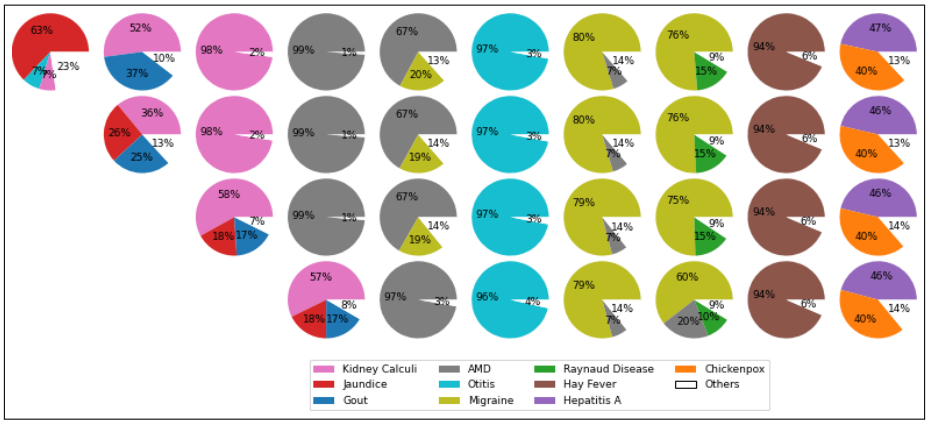


Figure 5: PUBMED10 topic distribution in the original dataset (left) and with EBCO co-clusterings (right; $g' \in [10..7]$).

The biomedical literature gives an explanation for this balanced association, as hepatitis has been found in several studies to be a severe complication of chickenpox in adult, with possibly lethal outcome. The association of these two topics corresponds to an actual biomedical fact, that is clearly reflected by the word partition. In Table 4 (last column), top terms contain *varicella*, *hepatitis*, *infection* and *outbreak*.

The Figure 5 also shows the gathering of several topics when g' is reduced. As an example, **Kidney Calculi** and **Gout** are associated when $g' = 10$, and are then associated with **Jaundice** when $g' \in [9, 8, 7]$. This topics gathering expresses a medical reality as Kidney stone disease is commonly seen in patients with underlying metabolic disorders such a Gout. Furthermore, Gout disease results in elevated levels of uric acid that can lead to crystals precipitating in the kidneys, inducing stone formation. In Table 4 (first column), the corresponding top terms are *uric*, *acid*, *oxalate*, *calcium*, *kidney* and *gout*. Several studies have demonstrated clinical associations between renal failure and obstructive jaundice – jaundice that results from abnormal retention of bile in the liver–. Specifically, the retention of bile constituents (eg. bilirubin) has deleterious effect on cardiovascular function, which in turn can induce kidney failure and tubular necrosis. As can be seen in Table 4, the top terms also contain *obstruction* and *jaundice*.

The Figure 5 also highlights the association between **Migraine** and **Raynaud's disease** for $g' \in [10..7]$, and with **AMD** when $g' = 7$. This again corresponds to biomedical facts. Specifically, Raynaud's disease patients hyper-react to phenomenom that constrict blood vessels (eg. cold, vibration, stress) leading to a lack of blood and oxygenation in digits and body parts. It has been established that Raynaud's is a frequent comorbid condition with Migraine. Some Migraine treatments are efficient on Raynaud's, such as calcium channel blockers. Yet, the most effective Raynaud's treatments, that are based on blood vessels dilatation are not recommended for migraineurs. Furthermore, studies have revealed genetic predisposition for Raynaud's disease, in particular an association with a polymorphism in the *NOS1* gene which is known for its role in cold induced vascular responses. Among the top terms from Table 4 (fifth column), *mutation*, *gene* or *polymorphism* clearly support this specificity of Raynaud's disease. These terms also strongly contrast with the top terms of the other co-cluster that has *Migraine* as main topic (Table 4, **Migraine** column). Finally, the constriction of blood vessels – frequently found in Raynaud's disease – that nourish the retina, is one of the *AMD* risk factors. This could explain the association of *AMD* to this co-cluster when $g' = 7$.

All in all, it appears that the co-clustering proposed by the fully unsupervised EBCO (with $g^\star = 7$) brings interesting biomedical indications on several actual disease relationships.

7 CONCLUSION

In unsupervised learning, *ensemble clustering* is a beneficial alternative to improve the quality of clustering. In our proposal, we have shown that this approach is undoubtedly useful to be extended in the context of text data co-clustering. Thereby, we have not only achieved this objective in terms of co-clustering quality and ease of co-clusters interpretation but we have also been able to address the crucial problem of choosing the number of co-clusters. This has been demonstrated on several datasets and compared to competitive co-clustering algorithms devoted to the same task. The EBCO is therefore tailored for document-term matrices and offers even a unified framework which can be exploited for other types of data. It is worth noting that the high time and space complexity of CoCE prevents handling large-scale data co-clustering; its computational cost is dominated by the execution of SVD on the n by d adjacency matrix. By contrast, EBCO uncovers an equivalence relationship with weighted double spherical k-means that dramatically decreases the time and space complexity to roughly linear complexity.

On the other hand, we have shown how our approach can be converted to an ensemble clustering approach. Furthermore, in terms of algorithmic or criterion, we established interesting connections with NMTF, spectral clustering and double weighted spherical k-means with *conscience mechanism*. These connections open up new prospects for investigation for other types of datasets.

ACKNOWLEDGMENTS

This work was supported by a grant overseen by the French National Research Agency (ANR-19-CE23-0002). It also received the labelling of *Cap Digital* and *EuroBiomed* competitiveness clusters.

REFERENCES

[1] Melissa Ailem, François Role, and Mohamed Nadif. 2015. Co-clustering document-term matrices by direct maximization of graph modularity. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. 1807–1810.

[2] Melissa Ailem, François Role, and Mohamed Nadif. 2016. Graph modularity maximization as an effective method for co-clustering text data. *Knowledge-Based Systems* 109 (2016), 160–173.

[3] Muna Al-Razgan and Carlotta Domeniconi. 2006. Weighted clustering ensembles. In *Proceedings of the 2006 SIAM International Conference on Data Mining*. SIAM, 258–269.

[4] Yanhua Chen, Lijun Wang, Ming Dong, and Jing Hua. 2009. Exemplar-based visualization of large document corpus (infovis2009-1115). *IEEE Transactions on Visualization and Computer Graphics* 15, 6 (2009), 1161–1168.

[5] Meghana Deodhar and Joydeep Ghosh. 2010. SCOAL: a framework for simultaneous co-clustering and learning from complex data. *ACM Transactions on Knowledge Discovery from Data* 4, 3 (2010), 11.

[6] Duane DeSieno. 1988. Adding a conscience to competitive learning. In *IEEE International Conference on Neural Networks*. 117–124.

[7] 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.

[8] Inderjit S Dhillon, Subramanyam Mallela, and Dharmendra S Modha. 2003. Information-theoretic co-clustering. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. 89–98.

[9] Xiaoli Zhang Fern and Carla E Brodley. 2004. Solving cluster ensemble problems by bipartite graph partitioning. In *Proceedings of the twenty-first international conference on Machine learning*. 36.

[10] Ana LN Fred and Anil K Jain. 2005. Combining multiple clusterings using evidence accumulation. *IEEE transactions on pattern analysis and machine intelligence* 27, 6 (2005), 835–850.

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

[12] 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.

[13] Blaise Hanczar and Mohamed Nadif. 2011. Using the bagging approach for biclustering of gene expression data. *Neurocomputing* 74, 10 (2011), 1595–1605.

[14] Blaise Hanczar and Mohamed Nadif. 2012. Ensemble methods for biclustering tasks. *Pattern Recognition* 45, 11 (2012), 3938–3949.

[15] Shudong Huang, Hongjun Wang, Dingcheng Li, Yan Yang, and Tianrui Li. 2015. Spectral co-clustering ensemble. *Knowledge-Based Systems* 84 (2015), 46–55.

[16] Lawrence Hubert and Phipps Arabie. 1985. Comparing partitions. *Journal of classification* 2, 1 (1985), 193–218.

[17] Lazhar Labiod and Mohamed Nadif. 2011. Co-clustering for binary and categorical data with maximum modularity. In *2011 IEEE 11th International Conference on Data Mining*. IEEE, 1140–1145.

[18] Tao Li and Chris Ding. 2008. Weighted consensus clustering. In *Proceedings of the 2008 SIAM International Conference on Data Mining*. SIAM, 798–809.

[19] Tao Li, Chris Ding, and Michael I Jordan. 2007. Solving consensus and semi-supervised clustering problems using nonnegative matrix factorization. In *Seventh IEEE International Conference on Data Mining (ICDM 2007)*. IEEE, 577–582.

[20] Hongfu Liu, Tongliang Liu, Junjie Wu, Dacheng Tao, and Yun Fu. 2015. Spectral ensemble clustering. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. 715–724.

[21] Sara C Madeira and Arlindo L Oliveira. 2004. Biclustering algorithms for biological data analysis: a survey. *IEEE/ACM TCBB* 1, 1 (2004), 24–45.

[22] 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.

[23] Yulong Pei, Nilanjan Chakraborty, and Katia Sycara. 2015. Nonnegative matrix tri-factorization with graph regularization for community detection in social networks. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

[24] Michael Röder, Andreas Both, and Alexander Hinneburg. 2015. Exploring the space of topic coherence measures. In *Proceedings of the eighth ACM international conference on Web search and data mining*. Shanghai, China, 399–408.

[25] François Role, Stanislas Morbieu, and Mohamed Nadif. 2019. CoClust: A Python Package for Co-Clustering. *Journal of Statistical Software* 88, 7 (2019), 1–29.

[26] 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.

[27] Aghiles Salah and Mohamed Nadif. 2019. Directional co-clustering. *Adv. Data Analysis and Classification* 13, 3 (2019), 591–620.

[28] Alexander Strehl and Joydeep Ghosh. 2003. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *The Journal of Machine Learning Research* 3 (2003), 583–617.

[29] Zhiqiang Tao, Hongfu Liu, Sheng Li, and Yun Fu. 2016. Robust spectral ensemble clustering. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. 367–376.

[30] Alexander Topchy, Anil K Jain, and William Punch. 2003. Combining multiple weak clusterings. In *Third IEEE International Conference on Data Mining*. IEEE, 331–338.

[31] Alexander Topchy, Anil K Jain, and William Punch. 2004. A mixture model for clustering ensembles. In *Proceedings of the 2004 SIAM international conference on data mining*. SIAM, 379–390.

[32] Iven Van Mechelen, Hans-Hermann Bock, and Paul De Boeck. 2004. Two-mode clustering methods: a structured overview. *Statistical Methods in Medical Research* 13, 5 (2004), 363–394.

[33] Sandro Vega-Pons, Jyrko Correa-Morris, and José Ruiz-Shulcloper. 2010. Weighted partition consensus via kernels. *Pattern Recognition* 43, 8 (2010), 2712–2724.

[34] Sandro Vega-Pons and José Ruiz-Shulcloper. 2011. A survey of clustering ensemble algorithms. *International Journal of Pattern Recognition and Artificial Intelligence* 25, 03 (2011), 337–372.

[35] Fei Wang, Xin Wang, and Tao Li. 2009. Generalized cluster aggregation. In *Twenty-First International Joint Conference on Artificial Intelligence*.

[36] Hua Wang, Feiping Nie, Heng Huang, and Chris Ding. 2011. Nonnegative matrix tri-factorization based high-order co-clustering and its fast implementation. In *2011 IEEE 11th international conference on data mining*. IEEE, 774–783.

[37] Hua Wang, Feiping Nie, Heng Huang, and Fillia Makedon. 2011. Fast nonnegative matrix tri-factorization for large-scale data co-clustering. In *Twenty-Second International Joint Conference on Artificial Intelligence*.

[38] Hye-Sung Yoon, Sun-Young Ahn, Sang-Ho Lee, Sung-Bum Cho, and Ju Han Kim. 2006. Heterogeneous clustering ensemble method for combining different cluster results. In *International Workshop on Data Mining for Biomedical Applications*. Springer, 82–92.

[39] Xianxue Yu, Guoxian Yu, Jun Wang, and Carlotta Domeniconi. 2019. Co-clustering Ensembles based on Multiple Relevance Measures. *IEEE Transactions on Knowledge and Data Engineering* (2019).