Machine Learning (2022) 111:1739–1764
https://doi.org/10.1007/s10994-022-06137-4

Semi-supervised Latent Block Model with pairwise constraints

Paul Riverain[1,2] · Simon Fossier[2] · Mohamed Nadif[1]

Received: 2 May 2021 / Revised: 25 January 2022 / Accepted: 6 February 2022 /
Published online: 16 March 2022

Abstract
Co-clustering aims at simultaneously partitioning both dimensions of a data matrix. It has demonstrated better performances than one-sided clustering for high-dimensional data. The Latent Block Model (LBM) is a probabilistic model for co-clustering based on mixture models that has proven useful for a broad class of data. In this paper, we propose to leverage prior knowledge in the form of pairwise semi-supervision in both row and column space to improve the clustering performances of the algorithms derived from this model. We present a general probabilistic framework for incorporating must link and cannot link relationships in the LBM based on Hidden Markov Random Fields. We instantiate this framework on a model for count data and present two inference algorithms based on Variational and Classification EM. Extensive experiments on simulated data and on real-world attributed networks confirm the interest of our approach and demonstrate the effectiveness of our algorithms.

Keywords Co-clustering · Latent Block Model · Semi-supervised Learning · Hidden Markov Random Fields

1 Introduction

Co-clustering, referred to by a variety of different names, aims at simultaneously partitioning both dimensions of a data matrix (Madeira & Oliveira, 2004; Van Mechelen et al., 2004; Govaert & Nadif, 2013). It has demonstrated better performances than one-sided

Communicated by Annalisa Appice, Sergio Escalera, Jose A. Gamez, Heike Trautmann.

✉ Paul Riverain
paul.riverain@etu.u-paris.fr

Simon Fossier
simon.fossier@thalesgroup.com

Mohamed Nadif
mohamed.nadif@u-paris.fr

[1] Centre Borelli UMR 9010, Université de Paris, 45 rue des Saints Pères, 75006 Paris, France

[2] Thales Research and Technology France, 1 Avenue Augustin Fresnel, 91120 Palaiseau, France

clustering for high-dimensional data (Banerjee et al., 2004; Deodhar & Ghosh, 2010; Yu et al., 2019; Bock, 2020; Affeldt et al., 2021). There are different approaches dedicated to co-clustering. Among these, the Latent Block Model (LBM) (Govaert & Nadif, 2005, 2018) is a probabilistic model devoted to co-clustering that allows to model data of different types using the appropriate mixture distribution (Govaert & Nadif, 2013; Ailem et al., 2017), to derive efficient co-clustering algorithms based on variants of the EM algorithm (Dempster et al., 1977) and to do model selection in order to determine the appropriate number of row and column clusters.

Semi-supervised (or constrained) clustering (Pensa & Boulicaut, 2008; Basu et al., 2008) has allowed clustering algorithms to better recover the clusters of a dataset with partial supervision on the set of data points. Co-clustering algorithms can benefit from side information in both row and column space (Song et al., 2010; Salah & Nadif, 2017; Salah et al., 2018; Affeldt et al., 2021). However, the existing semi-supervised approaches have not been presented to the probabilistic setting of the LBM.

In this paper, we propose a general model, namely HLBM, to incorporate side information in the form of pairwise constraints between the rows and the columns of a data matrix in the LBM. This semi-supervision is formulated in a probabilistic setup using Hidden Markov Random Fields (HMRF). We instantiate this model for count data with a Poisson distribution and propose two algorithms based on Classification EM and Variational EM. We analyze the behavior of these algorithms when varying the trade-off between the semi-supervision and the data likelihood on data simulated with the model. We apply our algorithm on real-world attributed networks and compare its clustering performances to existing algorithms.

2 Related work

In the domain of unsupervised image segmentation, Ambroise and Govaert (1998) propose to introduce similarity constraints between the data points in EM by optimizing a penalized variational criterion for one-sided clustering and investigate the convergence properties of their algorithm. Celeux et al. (2003) present three algorithms for clustering with mixture models and HMRF: mean, mode and simulated field EM and compare their algorithms to the iterated conditional modes algorithm (ICM) (Besag, 1986) that maximizes the pseudo-likelihood using a *Maximum A Posteriori* classification rule. The authors report the good performances of mean and simulated field EM algorithms compared with ICM.

Wagstaff et al. (2001) propose to add Must Link (ML) and Cannot Link (CL) constraints to the k-means algorithm. These constraints can not be violated and are not presented in a probabilistic context. Basu et al. (2004) presents a probabilistic framework that uses a HMRF to include ML and CL relationships and propose an algorithm based on ICM. This algorithm is compared with Belief Propagation and linear programming relaxation of the objective function in (Bilenko & Basu, 2004), where the authors present empirical evidence that ICM gives similar results as these more complex algorithms when the number of constraints is great enough. Lange et al. (2005) experimentally show that their algorithm based on deterministic annealing with mean-field variational inference generally gives better clustering performances than the ICM-based algorithm of Basu et al. (2004) even with a great number of constraints. Tang et al. (2009) considers ML and CL relationships in the context of graph clustering with matrix factorization by adding penalty terms computed with the Euclidean distance between the learned factors.

Pensa and Boulicaut (2008) present a constrained co-clustering algorithm based on a metric approach that includes ML, CL as well as interval constraints. These introduced constraints cannot be violated like in Wagstaff et al. (2001). The interval constraint is defined based on an ordering on the set of rows (or columns) and define an interval constraint on the set of row (or column) clusters such that the obtained one-sided clusters are intervals w.r.t. the given ordering. Kilic et al. (2016) propose a semi-supervised co-clustering algorithm in a "fuzzy" context in which the supervision is expressed by using fixed labels during inference for data points whose cluster is known. This approach is also used in Nam et al. (2020) in the context of the LBM. Yan et al. (2013) introduce "fuzzy" ML and CL relationships in their metric-based approach for co-clustering. The additive penalty terms introduced by the constraints in their objective functions (without justification) can be seen as a special case of ours where all relationships have the same weight.

Another approach to incorporating constraints in an unsupervised setting is based on manifold learning with Laplacian regularization. It consists in building a k-nearest neighbors graph that describes the intrinsic geometry of the data. It has been proposed in Zhu and Lafferty (2005) and He et al. (2011) on Gaussian mixture models, but the E or M steps can not be expressed in closed form, requiring gradient methods or heuristics. This has been applied in Salah and Nadif (2017) on von-Mises Fisher mixture models, in the context of item recommendation, to incorporate constraints from a social network that connects the users.

As we focus on the co-clustering task, the closest work to ours is the CITTC model of Song et al. (2010), where the authors propose a constrained version of the information theoretic co-clustering (ITCC) model of Dhillon et al. (2003) by using two HMRFs. However, Govaert and Nadif (2018) proved that ITCC with the Kullback-Leibler divergence as the chosen Bregman divergence is equivalent to the Poisson LBM with equal mixture proportions. This hypothesis of equal mixture proportions makes it difficult for the algorithm to recover unbalanced clusters. Moreover, on an algorithmic point of view, ITCC does not benefit from reduced intermediate matrices (see Sect. 4.1). As presented in Appendix 1, CITTC can be seen as a particular case of our model.

3 The proposed model

The data is represented by a matrix $\boldsymbol{X} = (x_{ij})$ of size $n \times d$, where the x_{ij} are assumed to be sampled from a given parametric distribution of density ϕ. The value of each entry of the data matrix depends on the latent row and columns partitions and on the model parameters.

3.1 Definition of the model

3.1.1 Sampling the latent variables in the HMRF

The partition of the set of rows in g clusters is represented by the latent classification matrix $\boldsymbol{Z} = (z_{ik})$, with $\sum_{k=1}^{g} z_{ik} = 1$, where $z_{ik} = 1$ if row i belongs to row cluster k and $z_{ik} = 0$ otherwise. Alternatively, we write $z_i \in \{1, \dots, g\}$ to be the cluster index of i. Similarly, the partition of the set of columns in m clusters is represented by the classification matrix $\boldsymbol{W} = (w_{j\ell})$, where $w_{j\ell} = 1$ or $w_j = \ell$ if column j belongs to column cluster ℓ. Denoting by $\mathcal{Z}$ and $\mathcal{W}$ the set of possible partitions of the rows and columns of $\boldsymbol{X}$ into respectively g and

m clusters, the latent space of the model is $\mathcal{Z} \times \mathcal{W}$. Let $\boldsymbol{\Theta}$ be the vector of parameters of the model. The classification matrices of the rows and the columns are a priori independent:

$$p(\boldsymbol{Z}, \boldsymbol{W}; \boldsymbol{\Theta}) = p(\boldsymbol{Z}; \boldsymbol{\Theta}) p(\boldsymbol{W}; \boldsymbol{\Theta}).$$

In the following, the semi-supervision over the set of rows (resp. columns) is expressed using pairwise and symmetric relationships between the latent classification vectors $\boldsymbol{z}_i$ (resp. $\boldsymbol{w}_j$). Let Υ_r (resp. Υ_c) be the set of rows (resp. columns) that are in a semi-supervision relationship. We write $\mathcal{N}_r(i) \subset \Upsilon_r$ to be the set of neighbors of row i, where $i \notin \mathcal{N}_r(i)$ and define a Markov Random Field (MRF) on Υ_r, where for $i \in \Upsilon_r$, the random variable z_i is dependent on a set of neighboring random variables $\{z_{i'} \mid i' \in \mathcal{N}_r(i)\}$. With MRF, the graph of conditional independence is undirected, that is $i' \in \mathcal{N}_r(i) \iff i \in \mathcal{N}_r(i')$. The Hammersley-Clifford theorem implies that the joint distribution of the MRF can be represented as a product of factors, one per maximal clique of the graph. In the following, we restrict the model to be a pairwise MRF, that is, the parameterization of the joint distribution is restricted to the edges of the graph, rather than the maximal cliques. For $i \in \Upsilon_r$ and $i' \in \mathcal{N}_r(i)$, we define the edge potential functions $\psi^r_{ii'}$ depending on the latent classification matrix $\boldsymbol{Z}$ and the potential parameter Ξ^r. For the other nodes, $i \notin \Upsilon_r$, we let the latent variables be independent random variables following a categorical distribution of parameter $\boldsymbol{\alpha}$, where $\sum_k \alpha_k = 1$, as in a classical LBM (Govaert & Nadif, 2008). Thus, the joint distribution over $\mathcal{Z}$ is given by:

$$p(\boldsymbol{Z}; \boldsymbol{\Theta}) = \Gamma_r(\Xi^r)^{-1} \exp\Bigg(\sum_{i \notin \Upsilon_r} \log \alpha_{z_i} + \frac{1}{2} \sum_{i \in \Upsilon_r} \sum_{i' \in \mathcal{N}_r(i)} \log \psi^r_{ii'}(z_i, z_{i'}; \Xi^r) \Bigg), \tag{1}$$

where Γ_r is the partition function of the HMRF on the rows. We can show that Γ_r only depends on Ξ^r (see Appendix 1). We define a similar HMRF on Υ_c for the columns, with potentials $\psi^c_{jj'}$, mixture proportions $\boldsymbol{\beta}$ and partition function $\Gamma_c(\Xi^c)$.

In the following, we consider two types of relationships: ML and CL relationships[1] and define the edge potential in the MRF so that nodes in a ML relationship are more likely to be in the same cluster and nodes in a CL relationship are more likely to be in different clusters. In order to define the potential functions, we consider a given symmetric weight matrix $\Xi^r = (\xi^r_{ii'})$, where $\xi^r_{ii'} \geq 0$ corresponds to the weight of the ML or CL relationship between row i and row i':

$$\log \psi^r_{ii'}(z_i, z_{i'}; \Xi^r) = \begin{cases} -\xi^r_{ii'} \mathbb{1}(z_{i'} \neq z_i) & (i, i') \in \mathcal{M}^r \\ -\xi^r_{ii'} \mathbb{1}(z_{i'} = z_i) & (i, i') \in \mathcal{C}^r. \end{cases}$$

where $\mathcal{M}^r$ (resp. $\mathcal{C}^r$) denotes the set of undirected edges representing a ML (resp. CL) relationship and $\mathbb{1}(.)$ returns 1 if its argument is true and 0 otherwise. In the same way, we define the edge potential for the set of columns $\psi^c_{jj'}(\boldsymbol{w}_j, \boldsymbol{w}_{j'}; \Xi^c)$, with parameter matrix $\Xi^c = (\xi^c_{jj'})$. This defines a distribution on the latent space.

[1] Note that, in our context, the semi-supervision is introduced in a probabilistic setting. Thus, the relationships could be called Should Link and Should not Link. However, we keep the names ML and CL for consistency with the existing literature (e.g. Basu et al. (2004))

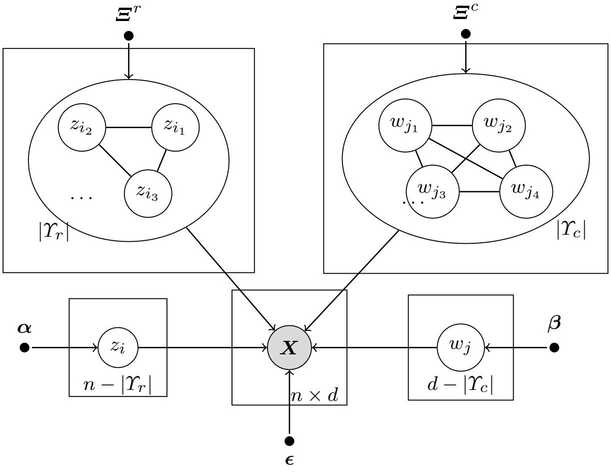


Fig. 1 Graphical model of the HLBM, where $\boldsymbol{\epsilon}$ is the parameter of the mixture distribution, $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_g)$ and $\boldsymbol{\beta} = (\beta_1, \dots, \beta_m)$ are the mixture proportions, Ξ^r and Ξ^c are the parameter matrices of the potentials of the HMRF

3.1.2 Sampling the observed variables

The univariate random variables x_{ij} are conditionally independent given $\boldsymbol{Z}$ and $\boldsymbol{W}$ and follow a probability distribution of density function ϕ and parameter $\boldsymbol{\epsilon} = (\epsilon_{ijk\ell})$: $x_{ij}|(z_{ik} = 1, w_{j\ell} = 1) \sim \phi(.;\epsilon_{ijk\ell})$. Note that the general definition with parameter $\epsilon_{ijk\ell}$ includes the more classical parameterization $\epsilon_{ijk\ell} = \gamma_{k\ell}$ and is not intended for practical use as such. The graphical model is depicted in Fig. 1. Thus,

$$p(\boldsymbol{X}|\boldsymbol{Z}, \boldsymbol{W};\boldsymbol{\Theta}) = \prod_{ij} \phi(x_{ij};\epsilon_{ijz_iw_j}) = \prod_{ijk\ell} \phi(x_{ij};\epsilon_{ijk\ell})^{z_{ik}w_{j\ell}}. \tag{2}$$

3.1.3 Complete data log-likelihood

The vector of parameters of the model is $\boldsymbol{\Theta} = \{\boldsymbol{\alpha}, \boldsymbol{\beta}, \epsilon, \Xi^r, \Xi^c\}$. Using (1) and (2), the complete data log-likelihood is given by:

$$\begin{aligned}
\log p(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{W};\boldsymbol{\Theta}) &= \log p(\boldsymbol{Z};\boldsymbol{\Theta}) + \log p(\boldsymbol{W};\boldsymbol{\Theta}) + \log p(\boldsymbol{X}|\boldsymbol{Z}, \boldsymbol{W};\boldsymbol{\Theta}) \\
&= \sum_{i \notin Y_r} \sum_k z_{ik} \log \alpha_k + \sum_{j \notin Y_c} \sum_\ell w_{j\ell} \log \beta_\ell \\
&\quad - \sum_{(i,i') \in \mathcal{M}^r} \xi^r_{ii'} \mathbb{1}(z_{i'} \neq z_i) - \sum_{(i,i') \in \mathcal{C}^r} \xi^r_{ii'} \mathbb{1}(z_{i'} = z_i) \\
&\quad - \sum_{(j,j') \in \mathcal{M}^c} \xi^c_{jj'} \mathbb{1}(w_{j'} \neq w_j) - \sum_{(j,j') \in \mathcal{C}^c} \xi^c_{jj'} \mathbb{1}(w_{j'} = w_j) \\
&\quad + \sum_{ijk\ell} z_{ik} w_{j\ell} \log \phi(x_{ij};\epsilon_{ijk\ell}) - \log \left(\Gamma_r(\Xi^r)\Gamma_c(\Xi^c)\right).
\end{aligned}$$

In this semi-supervised setup, $\boldsymbol{\Theta}$ can be decomposed as $\boldsymbol{\Theta} = \{\boldsymbol{\Theta}_L, \boldsymbol{\Theta}_F\}$, where $\boldsymbol{\Theta}_L = \{\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\epsilon}\}$ is to be learned and $\boldsymbol{\Theta}_F = \{\Xi^r, \Xi^c\}$ is fixed since it is given as input of the algorithms. In the following, we define $\boldsymbol{S}^r = (s^r_{ii'})$ such that:

$$s^r_{ii'} = \begin{cases} \lambda_r^{-1}\xi^r_{ii'} & (i,i') \in \mathcal{M}^r \\ -\lambda_r^{-1}\xi^r_{ii'} & (i,i') \in \mathcal{C}^r \\ 0 & \text{otherwise,} \end{cases}$$

where $\lambda_r > 0$ is a scaling factor for all the weights. The model is not identifiable for λ_r and $\boldsymbol{S}^r$, but this is not a problem since these parameters are fixed. Thereby, the log-potential can be simply written up to a constant as $\log \psi^r_{ii'}(z_i, z_{i'}; \lambda_r, \boldsymbol{S}^r) = \lambda_r s^r_{ii'} \sum_k z_{ik} z_{i'k}$. Similarly, we can define $\boldsymbol{S}^c = (s^c_{jj'})$ with scaling factor λ_c for the weights on the column space. Since Γ_r (resp. Γ_c) only depends only on Ξ^r (resp. Ξ^c) (see Appendix 1), the complete data log-likelihood is then reduced, up to a constant, to:

$$\begin{aligned} \log p(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{W}; \boldsymbol{\Theta}) = & \sum_{i \notin Y_r} \sum_k z_{ik} \log \alpha_k + \sum_{j \notin Y_c} \sum_\ell w_{j\ell} \log \beta_\ell \\ & + \frac{\lambda_r}{2} \sum_{ii'k} s^r_{ii'} z_{ik} z_{i'k} + \frac{\lambda_c}{2} \sum_{jj'\ell} s^c_{jj'} w_{j\ell} w_{j'\ell} + \sum_{ijk\ell} z_{ik} w_{j\ell} \log \phi(x_{ij}; \epsilon_{ijk\ell}) + C. \end{aligned} \tag{3}$$

3.1.4 Including an external field in the HMRF

In the proposed model, the rows or columns in the HMRFs do not contribute to the mixture proportions of the model. As proposed in (Celeux et al. 2002), we can address this problem by setting all nodes in the MRFs and defining mixture-like parameters $\boldsymbol{\alpha}$ as an external field, using node potentials. Thus, we can define the following variant of the model, where $Y_r = \{1, \dots, n\}$ and the joint distribution writes:

$$\log p(\boldsymbol{Z}; \boldsymbol{\Theta}) = -\log \Gamma_r(\boldsymbol{\alpha}, \lambda_r, \boldsymbol{S}^r) + \sum_{ik} z_{ik} \log \alpha_k + \frac{\lambda_r}{2} \sum_{ii'k} s^r_{ii'} z_{ik} z_{i'k}.$$

The specificity of this model is that the nodes that are in a semi-supervision relationship also contribute to the mixture-like parameter $\boldsymbol{\alpha}$. It must however be noted that the normalization constant Γ_r now depends on $\boldsymbol{\alpha}$, λ_r and $\boldsymbol{S}^r$.

3.2 Inference with the EM algorithm

In order to use the model in a clustering setting, we want to jointly infer the latent variables $\boldsymbol{Z}, \boldsymbol{W}$ and to learn the model parameters. In the following, we develop two approaches, based respectively on Classification EM (Celeux & Govaert, 1992) and on Variational EM (Govaert & Nadif, 2005).

3.2.1 Classification EM approach

In the Classification EM (CEM) approach (Govaert & Nadif, 2008), we maximize $\log p(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{W}; \boldsymbol{\Theta})$ by alternating the maximization w.r.t. $\boldsymbol{Z}$, $\boldsymbol{\Theta}$, $\boldsymbol{W}$ and $\boldsymbol{\Theta}$. However, here, the maximization w.r.t. a classification matrix, e.g. $\boldsymbol{Z}$ is not tractable because of the dependencies introduced in the HMRF. Thus, we use ICM, a coordinate ascent algorithm where we maximize $\log p(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{W};\boldsymbol{\Theta})$ w.r.t. $z_i \in \{1, \dots, g\}$, keeping $(z_{i'})_{i' \neq i}$ fixed. For $i \in Y_r$, the classification E-step is given by: $z_i = \operatorname{argmax}_{z_i} \log p(\boldsymbol{X}, z_i, (z_{i'})_{i' \neq i}, \boldsymbol{W};\boldsymbol{\Theta})$. Thus, we can show that the CE-step is:

$$z_i = \operatorname{argmax}_k \Big(\lambda_r \sum_{i'} s^r_{ii'} z_{i'k} + \sum_{j\ell} w_{j\ell} \log \phi(x_{ij};\epsilon_{ijk\ell})\Big).$$

The M-step is given by $\boldsymbol{\Theta} = \operatorname{argmax}_{\boldsymbol{\Theta}} \log p(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{W};\boldsymbol{\Theta})$ and depends on the class-conditional densities.

3.2.2 Variational EM approach

A variational approximation of the posterior distribution can be used as in (Govaert & Nadif, 2005). Let Q be a probability over the latent space $\mathcal{Z} \times \mathcal{W}$, parameterized by $\widetilde{\boldsymbol{Z}} = (\tilde{z}_{ik})$ and $\widetilde{\boldsymbol{W}} = (\tilde{w}_{j\ell})$, such that

$$Q(\boldsymbol{Z}, \boldsymbol{W};\widetilde{\boldsymbol{Z}}, \widetilde{\boldsymbol{W}}) = Q(\boldsymbol{Z};\widetilde{\boldsymbol{Z}})Q(\boldsymbol{W};\widetilde{\boldsymbol{W}}) = \prod_{ik} \tilde{z}_{ik}^{z_{ik}} \prod_{j\ell} \tilde{w}_{j\ell}^{w_{j\ell}},$$

and where $\sum_k \tilde{z}_{ik} = 1$. We can show that $Q(z_{ik} = 1) = \tilde{z}_{ik}$. The objective function $F(\widetilde{\boldsymbol{Z}}, \widetilde{\boldsymbol{W}}, \boldsymbol{\Theta})$ of the Variational EM algorithm is:

$$F(\widetilde{\boldsymbol{Z}}, \widetilde{\boldsymbol{W}}, \boldsymbol{\Theta}) = \mathbb{E}_Q\big(\log p(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{W};\boldsymbol{\Theta})\big) + H(Q).$$

The objective of this approach is to obtain $\widetilde{\boldsymbol{Z}}, \widetilde{\boldsymbol{W}}, \boldsymbol{\Theta} = \operatorname{argmax}_{\widetilde{\boldsymbol{Z}}, \widetilde{\boldsymbol{W}}, \boldsymbol{\Theta}} F(\widetilde{\boldsymbol{Z}}, \widetilde{\boldsymbol{W}}, \boldsymbol{\Theta})$, which can be reached by alternating the maximization of F wrt. $\widetilde{\boldsymbol{Z}}$, $\boldsymbol{\Theta}$, $\widetilde{\boldsymbol{W}}$, and $\boldsymbol{\Theta}$. Contrary to the classical LBM (Govaert & Nadif, 2008), the maximization of $F(\widetilde{\boldsymbol{Z}}, \widetilde{\boldsymbol{W}}, \boldsymbol{\Theta})$ wrt. $\widetilde{\boldsymbol{Z}}$ can not be decomposed for each row $i \in Y_r$ because of the dependencies introduced in the HMRF. However, we can apply coordinate ascent on $f(\tilde{z}_1, \dots, \tilde{z}_n) = F(\widetilde{\boldsymbol{Z}}, \widetilde{\boldsymbol{W}}, \boldsymbol{\Theta})$, by maximizing over $\tilde{z}_i = (\tilde{z}_{i1}, \dots, \tilde{z}_{ig})^\top$ and keeping fixed $(\tilde{z}_{i'})_{i' \neq i}$ such that the Lagrangian of each optimization problem is (see Appendix 1):

$$\begin{aligned}\mathcal{L}_\mu = {} & \frac{\lambda_r}{2} \sum_{i'} s^r_{i'i} \sum_k \tilde{z}_{i'k}\tilde{z}_{ik} + \frac{\lambda_r}{2} \sum_{i'} s^r_{ii'} \sum_k \tilde{z}_{ik}\tilde{z}_{i'k} \\ & + \sum_k \tilde{z}_{ik} \sum_{j\ell} \tilde{w}_{j\ell} \log \phi(x_{ij};\epsilon_{ijk\ell}) - \sum_k \tilde{z}_{ik} \log \tilde{z}_{ik} - \mu(1 - \sum_k \tilde{z}_{ik}).\end{aligned}$$

Thus, the VE-step is given by the following fixed-point:

$$\tilde{z}_{ik} \propto \begin{cases} \exp\big(\lambda_r \sum_{i'} s^r_{ii'} \tilde{z}_{i'k}\big) \prod_{j\ell} \phi(x_{ij};\epsilon_{ijk\ell})^{\tilde{w}_{j\ell}} & i \in Y_r \\ \alpha_k \prod_{j\ell} \phi(x_{ij};\epsilon_{ijk\ell})^{\tilde{w}_{j\ell}} & i \notin Y_r. \end{cases}$$

Note that we can deduce easily and in the same way the expression of $\tilde{w}_{j\ell}$. On the other hand, in the general formulation of Celeux et al. (2003), the proposed VE-steps corresponds to a mean field approximation followed by a regular E-step. The M-step is given by $\boldsymbol{\Theta} = \operatorname{argmax}_{\boldsymbol{\Theta}} F(\widetilde{\boldsymbol{Z}}, \widetilde{\boldsymbol{W}}, \boldsymbol{\Theta})$ and depends on the class-conditional densities.

3.2.3 With an external field

For a model with an external field, the E-steps for a row i are:

$$\begin{cases} \tilde{z}_{ik} \propto \alpha_k \exp\left(\lambda_r \sum_{i'} s^r_{ii'} \tilde{z}_{i'k}\right) \prod_{j\ell} \phi(x_{ij};\epsilon_{ijk\ell})^{\tilde{w}_{j\ell}} & \text{VE-step} \\ z_i = \operatorname{argmax}_k\left(\log \alpha_k + \lambda_r \sum_{i'} s^r_{ii} z_{i'k} + \sum_{j\ell} w_{j\ell} \log \phi(x_{ij};\epsilon_{ijk\ell})\right) & \text{CE-step.} \end{cases}$$

The new potential function includes the mixture parameters as a node potential, which makes the partition function dependent on the mixture parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. Thus, the M-step for the mixture parameters has no closed form. We propose to ignore this dependence for the M step. Experimentally, we did not observe any significant difference between the two versions of the model in terms of clustering performance, but this model offers easier computations. Consequently, we used this model in the rest of the paper.

3.3 The proposed algorithm

3.3.1 Iteration of the fixed point in the VE-step

The proposed E-step corresponds to a fixed point equation of the form $f(\widetilde{\boldsymbol{Z}}) = \widetilde{\boldsymbol{Z}}$. This is similar to the E-step of the Neighborhood EM algorithm of Ambroise and Govaert (1998) in the case of one-sided mixture models, where only ML relationships are considered. The authors prove that, for $\lambda_r < (\max_i \sum_{i'} |s^r_{ii'}|)^{-1}$, f is a contraction mapping and the corresponding fixed-point is the maximum of the objective criterion for the E-step. In our experiments, we observed that a single iteration of this fixed-point seems to suffice, which is also suggested in (Celeux et al., 2003). Contrary to (Ambroise & Govaert, 1998), we observed that this sufficient condition on λ_r was too restrictive in our case and could not lead to enough regularization (see Sect. 5).

3.3.2 Parallel updates

For efficiency reasons, we use parallel updates in the E-step for the VEM and CEM algorithms in our implementation. This procedure trades the convergence properties of the ICM algorithm (for CEM) or the fixed-point iteration (for VEM) for the benefit of parallel computations.

For VEM, as suggested in (Hinton et al., 2005), we use damping for the parallel updates in the VE-step in order to avoid oscillations. Let $\tilde{z}_{ik}^{(c+1/2)}$ be the variational probability obtained after one iteration of the fixed point in the E-step and let $\eta \in (0, 1)$. The damped VE-step is given by: $\tilde{z}_{ik}^{(c+1)} = (1-\eta)\tilde{z}_{ik}^{(c+1/2)} + \eta\tilde{z}_{ik}^{(c)}$. For CEM, we propose to use sequential updates after a given number of iterations if convergence has not been reached yet. The order of the updates is randomly selected at each E-step. Note that this choice of parallel updates does not allow the use of stochastic variants of EM based on variational inference as explained in (Celeux et al., 2003) for the simulated field algorithm.

3.4 Connection to other models

The proposed algorithms can be presented in connection to graph convolutional Neural Networks, where the constraints matrices are seen as the graph adjacency matrix, and to Laplacian regularization where the ML constraints can be viewed as a k-nearest neighbors graph describing the manifold on which the data lies. We here detail these connections.

3.4.1 Graph convolutional Neural Networks

Let $\Lambda_{ik}^{\widetilde{W}^{(c)}} = \prod_{j\ell} \phi(x_{ij};\epsilon_{ijk\ell}^{(c)})^{\tilde{w}_{j\ell}^{(c)}}$ (resp. $\Lambda_{j\ell}^{\widetilde{Z}^{(c)}} = \prod_{ik} \phi(x_{ij};\epsilon_{ijk\ell}^{(c)})^{\tilde{z}_{ik}^{(c)}}$) and $\boldsymbol{A}^{(c)}$ (resp. $\boldsymbol{B}^{(c)}$) be the $n \times g$ (resp. $d \times m$) matrix such that each row of the matrix is $\boldsymbol{\alpha}^{(c)}$ (resp. $\boldsymbol{\beta}^{(c)}$). The unnormalized variational probabilities at iteration c of the VEM algorithm can be written:

$$\begin{cases} \widetilde{\boldsymbol{Z}}_u^{(c+1)} & = \boldsymbol{A}^{(c)} \odot \exp\left(\lambda_r \boldsymbol{S}^r \widetilde{\boldsymbol{Z}}^{(c)}\right) \odot \boldsymbol{\Lambda}^{\widetilde{\boldsymbol{W}}^{(c)}} \\ \widetilde{\boldsymbol{W}}_u^{(c+1)} & = \boldsymbol{B}^{(c)} \odot \exp\left(\lambda_c \boldsymbol{S}^c \widetilde{\boldsymbol{W}}^{(c)}\right) \odot \boldsymbol{\Lambda}^{\widetilde{\boldsymbol{Z}}^{(c+1)}}. \end{cases}$$

This can be compared to the graph convolutional neural networks (GCN) of Kipf and Welling (2016a), in a supervised context, where the $(c+1)$th hidden layer $\boldsymbol{H}^{(c+1)}$ is given by $\boldsymbol{H}^{(c+1)} = \mathrm{ReLU}(\tilde{\boldsymbol{S}}\boldsymbol{H}^{(c)}\boldsymbol{\Omega}^{(c)})$, where $\tilde{\boldsymbol{S}} = \boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{S}_I\boldsymbol{D}^{-\frac{1}{2}}$ with $\boldsymbol{S}_I = \boldsymbol{S} + \boldsymbol{I}$, $\boldsymbol{D}$ is the diagonal degree matrix of $\boldsymbol{S}_I$, $\boldsymbol{S}$ is the adjacency matrix of the attributed graph, $\boldsymbol{H}^{(0)} = \boldsymbol{X}$ contains the attributes of the graph, and $\boldsymbol{\Omega}^{(c)}$ is the weight matrix of layer c. The node features are propagated through the nodes neighbors and at layer c, each node i has a latent representation $\boldsymbol{h}_i^{(c)}$ which aggregates the features of the nodes c steps away in the adjacency matrix $\boldsymbol{S}$. In our model, we do not propagate the nodes features through the nodes neighbors (in the observed graph) but we instead propagate the posterior probabilities through the nodes neighbors (in the HMRF). At iteration c of EM, each node has aggregated the posterior probabilities of nodes c steps away in the HMRF. For our model, the nodes features are modeled in the generative part of the E-step: $\boldsymbol{A} \odot \boldsymbol{\Lambda}^{\widetilde{W}^{(c)}}$ and the nodes aggregate their neighbors posterior probability with the kernel $\boldsymbol{S}^r$ (or $\boldsymbol{S}^c$), that is not learned contrary to GCNs (which requires a set of labeled data).

3.4.2 Laplacian regularization

A straightforward extension of Zhu and Lafferty (2005), He et al. (2011) and (Salah and Nadif 2017) in a semi-supervised context for the LBM is, given weighted adjacency matrices ($\boldsymbol{S}^r$ for the rows and $\boldsymbol{S}^c$ for the columns) that represent the ML relationships between the data points, to consider that two rows in a ML relationship have to lie close in the latent space. To this end, one can optimize a penalized log-likelihood $\log p(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{W}; \boldsymbol{\Theta}) - \lambda_r \mathcal{R}_r - \lambda_c \mathcal{R}_c$, where the penalty for rows is $\mathcal{R}_r = \frac{1}{2}\sum_{ii'}\sum_k s_{ii'}^r(\tilde{z}_{ik} - \tilde{z}_{i'k})^2 = \mathrm{Tr}\,(\widetilde{\boldsymbol{Z}}^\top \boldsymbol{L}^r \widetilde{\boldsymbol{Z}})$, where $\boldsymbol{L}^r$ is the Laplacian matrix associated to the adjacency matrix $\boldsymbol{S}^r$. As mentioned in (Zhu & Lafferty, 2005), $\mathcal{R}_r$ may seem to be a prior for the latent variables of the model in the form $p(\boldsymbol{Z};\boldsymbol{\Theta}) \propto \mathrm{Tr}\,(\widetilde{\boldsymbol{Z}}^\top \boldsymbol{L}^r \widetilde{\boldsymbol{Z}}) + \sum_{ik} z_{ik} \log \alpha_k$, but it actually depends on the posterior probabilities and is thus best thought of as a discriminative component in the objective function. In our model, HLBM, the semi-supervision is expressed in a generative way with the

HMRFs, but EM algorithms for the two models can be compared. In the CEM approach for the Laplacian regularization, we optimize the regularized complete data log-likelihood over the latent classification matrices and we can then consider $\tilde{z}_{ik} = z_{ik} \in \{0, 1\}$. Thus, $\mathcal{R}_r = \frac{1}{2} \sum_{ii'} s^r_{ii'} \mathbb{1}(z_{i'} \neq z_i)$. It can then be shown that the corresponding algorithm is equivalent to the algorithm of HLBM for CEM. In the VEM approach, due to the non-linearities in the latent variables introduced by the Laplacian regularization term, there is no closed form for the variational E-step. The solution proposed in (He et al., 2011; Salah & Nadif, 2017) is to maximize the variational objective and minimize the regularization term sequentially. This strategy, unfortunately, did not yield convincing results for our model.

4 Co-clustering of count data with the Poisson HLBM

With the appropriate mixture distribution, the proposed model can be applied on different types of data, as in classical mixture models: gaussian distributions can be chosen to model microarray data or multinomial distributions for categorical data. In the following, we develop the proposed model for count data with a mixture of Poisson distributions, as in (Govaert & Nadif, 2018). This model has the advantage of being suited to high-dimensional text data (Ailem et al., 2017).

4.1 Algorithm for the Poisson HLBM

The data distribution, conditionally on the clusters is $x_{ij} | z_{ik} w_{j\ell} = 1 \;\sim\; \phi(., \mu_i \nu_j \gamma_{k\ell})$, where ϕ is the probability mass function of a Poisson distribution. The model is parameterized by $\boldsymbol{\Theta} = \{\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\nu}, \boldsymbol{\gamma}, \boldsymbol{\Xi}^r, \boldsymbol{\Xi}^c\}$. For identifiability, following (Govaert & Nadif, 2018), we impose the following constraint: for $M > 0$,

$$\boldsymbol{\Theta} \in \{\boldsymbol{\Theta} | \mu_. = \nu_. = M, \forall k \sum_{\ell} \beta_\ell \gamma_{k\ell} = M^{-1}, \forall \ell \sum_k \alpha_k \gamma_{k\ell} = M^{-1}\}.$$

It can then be shown that $\mathbb{E}(x_{i.}) = \mu_i$ and $\mathbb{E}(x_{.j}) = \nu_j$, and the marginals μ_i and ν_j can then be replaced by $x_{i.} = \sum_j x_{ij}$ and $x_{.j} = \sum_i x_{ij}$. Thereby, we define respectively the row, column and block reduced matrices $\boldsymbol{X}^{\widetilde{\boldsymbol{Z}}} = (x^{\widetilde{\boldsymbol{Z}}}_{kj})$, $\boldsymbol{X}^{\widetilde{\boldsymbol{W}}} = (x^{\widetilde{\boldsymbol{W}}}_{i\ell})$ and $\boldsymbol{X}^{\widetilde{\boldsymbol{Z}}\widetilde{\boldsymbol{W}}} = (x^{\widetilde{\boldsymbol{Z}}\widetilde{\boldsymbol{W}}}_{k\ell})$ such that $\boldsymbol{X}^{\widetilde{\boldsymbol{Z}}} = \widetilde{\boldsymbol{Z}}^\top \boldsymbol{X}$, $\boldsymbol{X}^{\widetilde{\boldsymbol{W}}} = \boldsymbol{X}\widetilde{\boldsymbol{W}}$ and $\boldsymbol{X}^{\widetilde{\boldsymbol{Z}}\widetilde{\boldsymbol{W}}} = \widetilde{\boldsymbol{Z}}^\top \boldsymbol{X}\widetilde{\boldsymbol{W}}$. The matrices $\widetilde{\boldsymbol{Z}}$ and $\widetilde{\boldsymbol{W}}$ contain the variational probabilities, as defined in Sect. 3.2. Equivalent reduced matrices $\boldsymbol{X}^{\boldsymbol{Z}}$, $\boldsymbol{X}^{\boldsymbol{W}}$ and $\boldsymbol{X}^{\boldsymbol{Z}\boldsymbol{W}}$ can be defined in a CEM setup. We can show that the VE and CE-steps are respectively:

$$\begin{cases} \tilde{z}_{ik} \propto \alpha_k \exp\left(\lambda_r \sum_{i'} s^r_{ii'} \tilde{z}_{i'k} + \sum_\ell x^{\widetilde{\boldsymbol{W}}}_{i\ell} \log \gamma_{k\ell}\right) & \text{VE-step} \\ z_i = \operatorname{argmax}_k \left(\log \alpha_k + \lambda_r \sum_{i'} s^r_{ii'} z_{i'k} + \sum_\ell x^{\boldsymbol{W}}_{i\ell} \log \gamma_{k\ell} \right) & \text{CE-step} \end{cases}$$

Since the constraints matrices are sparse, these updates are efficient. The M-step in a VEM setting leads to: $\alpha_k = \frac{\sum_i \tilde{z}_{ik}}{n} = \frac{\tilde{z}_{.k}}{n}$, $\beta_\ell = \frac{\sum_j \tilde{w}_{j\ell}}{d} = \frac{\tilde{w}_{.\ell}}{d}$, $\gamma_{k\ell} = \frac{x^{\widetilde{\boldsymbol{Z}}\widetilde{\boldsymbol{W}}}_{k\ell}}{x^{\widetilde{\boldsymbol{Z}}}_{k.} x^{\widetilde{\boldsymbol{W}}}_{.\ell}}$. The M-step for CEM is similar, with hard assignments. Following (Govaert & Nadif, 2008), we propose the following algorithm for the Poisson HLBM VEM (`PHLBMVEM`) (see Algorithm 1). Note that the row and column M-step for $\boldsymbol{\gamma}$ can benefit from the reduced matrix $\boldsymbol{X}^{\boldsymbol{Z}}$ or $\boldsymbol{X}^{\boldsymbol{W}}$ computed for its corresponding E-step. The algorithm for CEM is similar, but considers hard assignment matrices, does not apply damping and switches to a sequential E-step after a given number of iterations.

```
Algorithm 1: PHLBMVEM
Input: Data matrix $\boldsymbol{X}$, constraints matrices $\boldsymbol{S}^r$ and $\boldsymbol{S}^c$, number of row and column
       clusters $g$ and $m$, damping factor $\eta$.
Output: Classification matrices $\boldsymbol{Z}, \boldsymbol{W}$, parameters $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}$
Initialization: Initialize $\widetilde{\boldsymbol{Z}}, \widetilde{\boldsymbol{W}}$ and set $\alpha_k = \frac{\tilde{z}_{.k}}{n}$, $\beta_\ell = \frac{\tilde{w}_{.\ell}}{d}$, $\gamma_{k\ell} = \frac{x_{k\ell}^{\widetilde{\boldsymbol{Z}}\widetilde{\boldsymbol{W}}}}{x_{k.}^{\widetilde{\boldsymbol{Z}}} x_{.\ell}^{\widetilde{\boldsymbol{W}}}}$
while Not converged do
    Compute $x_{i\ell}^{\widetilde{\boldsymbol{W}}} = \sum_j \tilde{w}_{j\ell} x_{ij}$;
    Row VE-step:
      – In parallel, $\forall i$, $\tilde{z}_{ik} \propto \alpha_k \exp\left(\lambda_r \sum_{i'} s_{ii'}^r \tilde{z}_{i'k} + \sum_\ell x_{i\ell}^{\widetilde{\boldsymbol{W}}} \log \gamma_{k\ell}\right)$;
      – Normalize $\widetilde{\boldsymbol{Z}}$ and apply damping;
    Row M-step: $\alpha_k = \frac{\tilde{z}_{.k}}{n}$, $\gamma_{k\ell} = \frac{\sum_i \tilde{z}_{ik} x_{i\ell}^{\widetilde{\boldsymbol{W}}}}{x_{.\ell}^{\widetilde{\boldsymbol{W}}} \sum_i \tilde{z}_{ik} x_{i.}}$;
    Compute $x_{kj}^{\widetilde{\boldsymbol{Z}}} = \sum_i \tilde{z}_{ik} x_{ij}$;
    Column VE-step:
      – In parallel, $\forall j$, $\tilde{w}_{j\ell} \propto \beta_\ell \exp\left(\lambda_c \sum_{j'} s_{jj'}^c \tilde{w}_{j'\ell} + \sum_k x_{kj}^{\widetilde{\boldsymbol{Z}}} \log \gamma_{k\ell}\right)$;
      – Normalize $\widetilde{\boldsymbol{W}}$ and apply damping;
    Column M-step: $\beta_\ell = \frac{\tilde{w}_{.\ell}}{d}$, $\gamma_{k\ell} = \frac{\sum_j \tilde{w}_{j\ell} x_{kj}^{\widetilde{\boldsymbol{Z}}}}{x_{k.}^{\widetilde{\boldsymbol{Z}}} \sum_j \tilde{w}_{j\ell} x_{.j}}$;
end
```

4.1.1 Initialization

We can use the constraints matrices $\boldsymbol{S}^r$ and $\boldsymbol{S}^c$ to provide a better initialization of the row and column partitions. We build a stochastic matrix $\boldsymbol{M}^r = \boldsymbol{S}_I^{r+} \boldsymbol{\Delta}_{r+}$, where $\boldsymbol{S}_I^{r+} = \boldsymbol{S}^{r+} + \boldsymbol{I}$, $\boldsymbol{S}^{r+}$ contains only the non-negative values of $\boldsymbol{S}^r$, and $\boldsymbol{\Delta}_{r+}$ is the inverse of the diagonal degree matrix of $\boldsymbol{S}_I^{r+}$. This corresponds to averaging the features of neighboring nodes. We apply a clustering algorithm, namely `skmeans` for a Poisson model (see Sect. 4), on the rows of $\boldsymbol{M}^r\boldsymbol{X}$ to get an initial partition for $\boldsymbol{Z}$. We apply the same procedure for the columns, using $\boldsymbol{S}^c$ and apply a clustering algorithm on the columns of $\boldsymbol{X}\boldsymbol{M}_c$. The obtained matrix has values $(\boldsymbol{M}^r\boldsymbol{X})_{ij} = x_{ij} + \frac{1}{s_{i.}^{r+}} \sum_{i'} s_{ii'}^{r+} x_{i'j}$, where the *dot* indicates the sum over a given index ($s_{i.} = \sum_{i'} s_{ii'}$). This method also has the advantage of reducing the sparsity of the data matrix used for initialization and thus provides a better partition.

4.2 Algorithmic complexity

The proposed algorithms can benefit from the sparse structure of $\boldsymbol{X}$, $\boldsymbol{S}^r$ and $\boldsymbol{S}^c$. The computations in the E-steps and M-steps are based on reduced matrices $\boldsymbol{X}^{\boldsymbol{Z}} = \boldsymbol{Z}^\top \boldsymbol{X}$, $\boldsymbol{X}^{\boldsymbol{W}} = \boldsymbol{X}\boldsymbol{W}$ and $\boldsymbol{X}^{\boldsymbol{ZW}} = \boldsymbol{Z}^\top \boldsymbol{X}\boldsymbol{W}$ of respective sizes $g \times d$, $n \times m$ and $g \times m$. Let n_{it} denote the number of iterations of the EM algorithm, n_{NZX} and n_{NZS_r} denote respectively the number of non-zero values in $\boldsymbol{X}$ and $\boldsymbol{S}^r$ and $\mathcal{N}_r^{\max} = \max_i |\mathcal{N}_r(i)|$.

For an iteration of the VEM algorithm, the computational bottleneck is the row and column cluster assignments, which is $O(gn(\mathcal{N}_r^{\max} + m))$ for the rows and $O(md(\mathcal{N}_c^{\max} + g))$ for the columns, and the computation of the reduced matrices is $O(n_{\mathrm{NZX}}(g + m))$. The time complexity of the Poisson VEM algorithm is thus $O\big(n_{\mathrm{it}}\big(n_{\mathrm{NZX}}(g + m) + gn(\mathcal{N}_r^{\max} + m) + md(\mathcal{N}_c^{\max} + g)\big)\big)$. The Poisson CEM complexity is similar to VEM but the algorithm benefits from a faster convergence and sparse structures for the classification matrices that speed-up the computations.

The space complexity of the Poisson VEM algorithm is related to the data matrix, the constraint matrix, the reduced matrices and the classification matrices. It is thus $O(n_{\mathrm{NZX}} + n_{\mathrm{NZS}_r} + n_{\mathrm{NZS}_c} + nm + dg + nd + dm)$. Sparse structures for CEM can reduce it to $O(n_{\mathrm{NZX}} + n_{\mathrm{NZS}} + nm + dg + n + d)$.

5 Experiments on simulated data

5.1 Sampling and experiment plan

Here, we evaluate our algorithms in terms of co-clustering performance on simulated data. In a semi-supervised setting, the difficulty of the co-clustering problem will depend on both the data and the given constraints. In the following, we describe the procedures to sample different co-clustering problems from the model, to generate different constraints matrices from the true clusters and to evaluate the obtained partitions against the true clusters.

5.1.1 Sampling the data

We can use the generative part of the model to sample simulated data. To this end, we do not include the HMRF to sample the data, and, given a vector of parameters $\boldsymbol{\Theta}$ we can sample the complete data $(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{W})$. The margins μ_i and ν_j are sampled from $\{1, \ldots, 100\}$ with a power law $p(k) \propto k^{-\frac{3}{2}}$, resulting in skewed margins. The experiments are carried out with $n = 100$ rows, $d = 200$ columns, $g = 3$ row clusters and $m = 4$ column clusters and mixture proportions $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ with symmetric Dirichlet distribution of parameter $\delta = 4$ and for $\boldsymbol{\gamma} = \gamma^0 \begin{pmatrix} 1 & 2 & 3 & 1 \\ 3 & 1 & 2 & 3 \\ 2 & 3 & 1 & 3 \end{pmatrix}$. where $\gamma^0 > 0$ controls the class overlap. In order to obtain 3 sets of parameters with increasing overlap $\{\boldsymbol{\Theta}_{+}, \boldsymbol{\Theta}_{++}, \boldsymbol{\Theta}_{+++}\}$, we measure the linear separability of the clusters with Linear Discriminant Analysis, computed as the ratio between the inter-cluster variance and the total variance of the data projected onto each of the factorial axes. Each of the ratios is in $[0,\ 1]$, a ratio of 1 meaning that the intra-cluster variance on the factorial axis is null (i.e. the clusters are linearly separable), and a ratio of 0 meaning that the centers of gravity of each cluster are projected onto the same point on the factorial axis. Using this criterion, we define $\gamma^0_{+} = 2 \times 10^{-2}$, $\gamma^0_{++} = 2 \times 10^{-3}$ and $\gamma^0_{+++} = 1 \times 10^{-3}$.

5.1.2 Sampling the constraints matrix

For each set of complete data, we can build row and column binary constraints matrices with some of the true classes, by setting $s^r_{ii} = 1$ if $z_{i'} = z_i$ or $s^r_{ii} = -1$ if $z_{i'} \neq z_i$. To this end, we sample a fraction f_S of all the $\begin{pmatrix} n \\ 2 \end{pmatrix}$ or $\begin{pmatrix} d \\ 2 \end{pmatrix}$ pairwise ML and CL constraints that can be formulated from the true clusters. It must be noted that the models will be influenced differently depending on which relationships are sampled and that CL relationships are sampled more often than ML relationships (with respective probabilities $1 - \sum_k \alpha_k^2$ and $\sum_k \alpha_k^2$ for a relationship on the set of rows). Thus, for each set of complete data we sample 50 row and column constraints matrices. We use an identical regularization parameter for rows and columns, $\lambda = \lambda_r = \lambda_c$.

In order to evaluate the sensibility of the models w.r.t. noise in the constraints matrix, we sample a fraction f_{noise} of the pairs (i, i') and set $s_{ii'}^{\text{noise}} = -s_{ii'}$. For the other (i, i') pairs, we set $s_{ii'}^{\text{noise}} = s_{ii'}$. Finally, given a set of ML and CL relationships, we can choose whether to apply transitive closure on these relationships as described in (Basu et al., 2004). This allows to test if the algorithm applies the transitive closure implicitly or if these supplementary relationships convey new information.

5.1.3 Measuring the information in the constraints matrices

For a given algorithm, the ML and CL relationships in a constraints matrix can convey more or less information on a clustering problem. They can be of limited use for an algorithm which naturally recovers these constraints without supervision and they can be noisy and convey wrong information about the true clusters—and thus be contradictory with the data—so that some constraints are not satisfied after convergence of the algorithm with supervision.

For a partition $\mathcal{P}$ of the rows or columns of the data matrix, where $\mathcal{P}_i$ is the cluster of node i, and a constraints matrix $\boldsymbol{S}$, we define the ratio (4), where $\text{unsat}(\mathcal{P}_i, \mathcal{P}_{i'}, s_{ii'})$ equals 1 if the constraint $s_{ii'}$ is not satisfied and 0 if the constraint is satisfied or if $s_{ii'} = 0$.

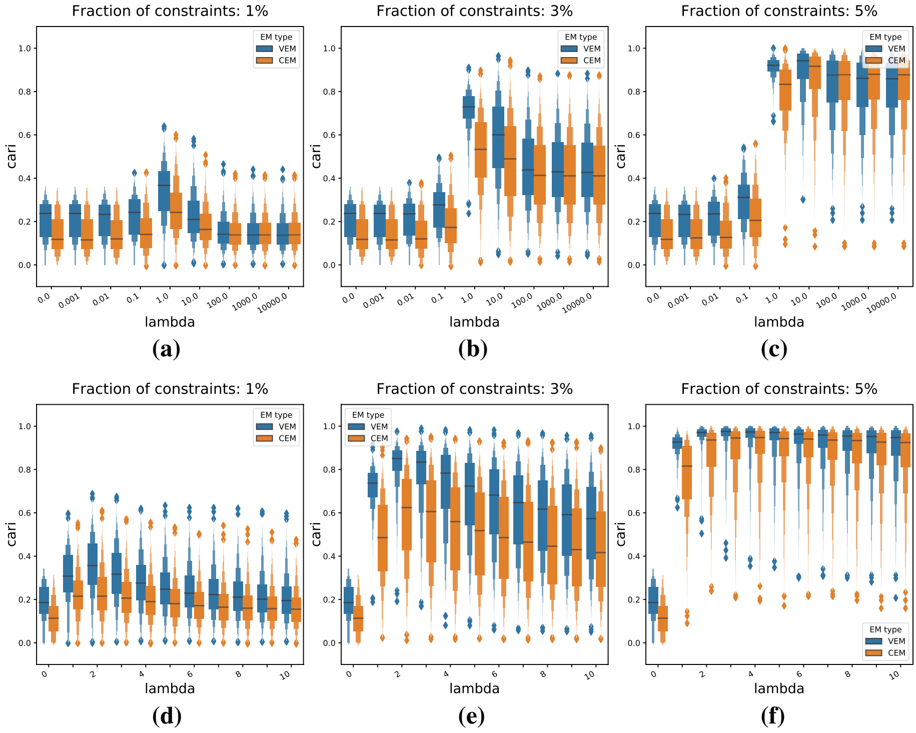
$$\mathcal{R}(\boldsymbol{S}, \mathcal{P}) = \frac{\sum_{ii'} |s_{ii'}| \text{unsat}(\mathcal{P}_i, \mathcal{P}_{i'}, s_{ii'})}{\sum_{ii'} |s_{ii'}|} \tag{4}$$

Depending on the nature of $\mathcal{P}$, this criterion can have different meanings. If $\mathcal{P}$ is a partition returned by the algorithm without regularization, $\mathcal{R}$ is the weighted proportion of constraints that are not already in the data. This criterion corresponds to a weighted version of the informativeness criterion of Davidson et al. (2006). If $\mathcal{P}$ is the true partition, $\mathcal{R}$ is a measure of noise in the constraints matrix and corresponds to a weighted version of the "spatial discordance" criterion of Miele et al. (2014). If $\mathcal{P}$ is the partition returned by the algorithm with regularization, $\mathcal{R}$ corresponds to the weighted proportion of constraints that have been violated by the algorithm after convergence. It can be noted that $\mathcal{R}$ is then proportional to the log prior for the rows or columns, in a setup without mixture proportions. This last criterion can also be used as an heuristic to suggest appropriate values for the regularization parameters λ_r and λ_c.

5.1.4 Experiment plan

We sample data from 3 sets of parameters with increasing overlap $\{\boldsymbol{\Theta}_+, \boldsymbol{\Theta}_{++}, \boldsymbol{\Theta}_{+++}\}$. For each $\boldsymbol{\Theta}$, we sample 50 sets of complete data $\{\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{W}\}$. For each set of complete data we initialize the row and column partitions $\boldsymbol{Z}^{(0)}$ and $\boldsymbol{W}^{(0)}$ by applying `skmeans` (50 initializations) on the sets of row and columns,[2] and for each $f_S \in \{0\%, 1\%, 2\%, 3\%, 4\%, 5\%\}$, we repeat the following procedure 20 times: sample the row and columns constraints matrices, then fit the model for each $\lambda \in \{0, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10^1, 10^2, 10^3, 10^4\}$. We report the classification results in terms of Co-clustering ARI (CARI) (Robert et al., 2021) between the true partitions and the partitions returned by our algorithm for each run of the algorithm. This criterion is an extension of the Adjusted Rand Index (Hubert & Arabie, 1985)

[2] We do not use the initialization described in 4.1.1 in order to focus on the model's ability to benefit from the semi-supervision.



(a) **(b)** **(c)**

(d) **(e)** **(f)**

Fig. 2 Comparison of CARI values for CEM and VEM algorithms and for different values of the regularization parameter λ. The results are presented for different fractions of ML and CL constraints sampled. The data is sampled with $\boldsymbol{\Theta}_{+++}$. *First row*: λ in log-scale, *second row*: λ in linear scale around a working point. *From left to right*: Increasing fraction f_S of sampled relationships

in the context of co-clustering. The CARI index varies between 0 and 1, where 1 means a perfect match and 0 leads to all worst scenarios, including independence and is proportional to the number of miss-classified cells in the data matrix. We identify a working point between two values of λ in the log-scale and also report the CARI values in a linear scale between these two values. In our experiments, we use parallel E-steps with a damping coefficient $\eta = 0.7$ in the case of VE-steps. We set the number of row and column cluster to their true value.

5.2 Evaluation of the CEM and VEM algorithms in terms of co-clustering

In Fig. 2, we compare, for a difficult clustering problem $\boldsymbol{\Theta}_{+++}$, the CARI values for the CEM and VEM algorithms in the absence of noise, as a function of f_S and λ, points with $\lambda = 0$ being our baseline without regularization. First, we observe that the regularization is almost always beneficial to the clustering performances. Also, we note that VEM gives better CARI values than CEM. We observe that the performances are sensitive to the choice of λ. In our experimental setup, the optimal working point of our algorithms is located between 1 and 10, independently of the class overlap. For easier problems $\boldsymbol{\Theta}_{+}$ and $\boldsymbol{\Theta}_{++}$, we observed, as expected, that the fraction of constraints f_S required to reach CARI ~ 1 increases with the complexity of the clustering problem.

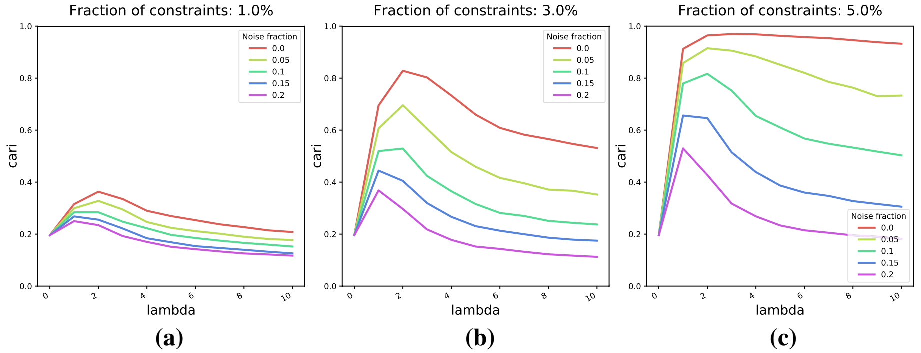


Fig. 3 Median CARI values for different values of noise in $\boldsymbol{S}^r$ and $\boldsymbol{S}^c$. The results are presented for different fractions of ML and CL constraints sampled and for different values of the regularization parameter λ. The data is sampled with $\boldsymbol{\Theta}_{+++}$. *From left to right*: Increasing fraction f_S of sampled relationships

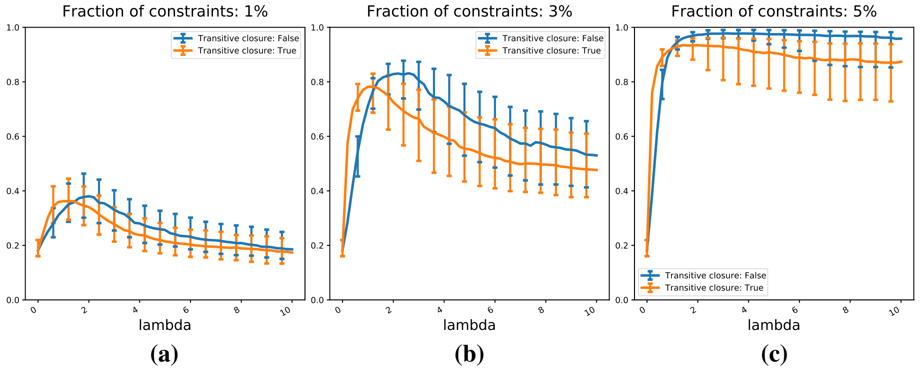


Fig. 4 Median CARI values with and without applying the transitive closure of the ML and CL relationships. Error bars correspond to 1st and 3rd quartiles. The results are presented for different fractions of ML and CL constraints sampled and for different values of the regularization parameter λ. The data is sampled with $\boldsymbol{\Theta}_{+++}$. *From left to right*: Increasing fraction f_S of sampled relationships

An interesting result is that the performances are more sensitive to the choice of λ for lower constraint fractions. When there is more supervision, this phenomenon is less prominent, and λ can more freely be set anywhere above a threshold value. Also, surprisingly, the CARI difference between VEM and CEM does not seem to increase with the fraction of constraints sampled (i.e. the density of the edges in the HMRF). It should however be noted that the drawbacks of CEM in terms of classification performance are counterbalanced by its fast convergence and its use of sparse data structures.

In Fig. 3, we observe that, when noise is added to the constraint matrices $\boldsymbol{S}^r$ and $\boldsymbol{S}^c$, the algorithm can still benefit from the regularization but the choice of λ becomes more critical and too high values of λ are detrimental to the clustering performances of the algorithm. As the fraction of true constraints increases, the algorithm becomes less dependent on the value of λ. In Fig. 4, we observe that applying the transitive closure of the ML and CL relationships does not seem to increase the clustering performances of the algorithm, as mentioned in (Lange et al., 2005). Moreover, it decreases the sparsity of the constraints matrices and thus the performances of the computations. Also, it increases the weight of

Table 1 Datasets characteristics

	Normalization $\boldsymbol{X}$	Type $\boldsymbol{A}$	n	d	g	$n_{\mathrm{NZX}}(\%)$	#Edges	Balance (%)
Cora	Binary	Binary	2708	1433	7	98.73	5294	22
Citeseer	Binary	Binary	3312	3703	6	99.14	4732	35
Wiki	tfidf	Weighted	2405	4973	17	86.46	17981	2
Pubmed	tfidf	Binary	19717	500	3	89.98	44338	52

the discriminative component w.r.t. the generative component and thus the value of λ must be changed accordingly (with a smaller value than without transitive closure). In practice, depending on which relationships are sampled, the constraints matrices after transitive closure will have different sparsity values and will consequently require different values of λ, which is not convenient. Finally, the results presented in Fig. 4 suggest that the transitive closure of the relationships is applied implicitly in the algorithm.

Thus, the regularization parameters λ_r and λ_c—which correspond to a scaling factor of the weights of the ML and CL relationships—must be set, as a first approximation, according to the confidence we have w.r.t. the given ML and CL relationships. The interval [0, 10] seems to be a suitable range for this parameter. It should however be noted that, even for ML and CL relationships sampled from the true clusters (i.e. without noise in $\boldsymbol{S}^r$ and $\boldsymbol{S}^c$), the algorithms remain sensitive—in a minor extent—to the choice of the regularization parameter, but are rarely affected negatively by the regularization. Finally, this dependency on the regularization parameters is reduced when the number of ML and CL constraints increases.

6 Experiments on real world data: attributed network clustering

In the previous Section, we evaluated our algorithms, in a semi-supervised setting, in terms of co-clustering performances on data sampled from the model. Here, we compare our algorithms to other algorithms from the literature in terms of one-sided clustering on real world data. We focus on the task of Attributed Network Clustering, where the data is in the form $(\boldsymbol{A}, \boldsymbol{X})$, where $\boldsymbol{A}$ is a graph adjacency matrix and $\boldsymbol{X}$ is a data matrix containing feature vector for each node in the network. We evaluate our algorithms on the task of clustering the nodes of these networks. Note that, several studies have demonstrated the importance of co-clustering even to obtain object clusters only (one-side clustering). Actually using co-clustering is often more effective than one-way clustering, especially when considering sparse high dimensional data.

6.1 Experimental setup

We evaluate the one-sided row clustering performances of the VEM and CEM algorithms on datasets commonly used in the field of Attributed Network Embedding (ANE). These datasets are four citation networks: Cora, Citeseer, Wiki and Pubmed where $\boldsymbol{A}$ is a sparse graph adjacency matrix in which each node corresponds to a document and edges correspond to citations, $\boldsymbol{X}$ is a data matrix containing a bag-of-words feature vector for each node in the network. Although our model is not directly suited for this kind of task, we can see the adjacency matrix $\boldsymbol{A}$ as a ML constraint matrix $\boldsymbol{S}^r$ on the set of rows. This is less expressive than graph convolutions where the model can learn complex aggregations of a node's neighbors features but this can still lead to satisfying results.

The datasets characteristics are reported on Table 1, where $n_{\text{NZX}}(\%)$ corresponds to the percentage of sparsity of the data matrix $\boldsymbol{X}$ and the balance coefficient is defined as the ratio of the number of documents in the smallest class to the number of documents in the largest class. The attribute matrices of the datasets are high-dimensional. Thus, co-clustering is an appropriate approach for these datasets.

Note that here, on binary data, we use the Poisson HLBM instead of a Bernoulli HLBM. It leads to give better results since the margins $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ implicitly perform a normalization of the data. This is relevant for bag-of-words data since the number of words in a document is not relevant in order to determine its cluster. Note also that the Poisson LBM is even used on Tf-idf normalized data (Wiki and Pubmed), which happens to give satisfying results.

6.2 Model selection

To assess the number of row and column clusters g and m, we rely on the asymptotic integrated classification likelihood (ICL) (Biernacki et al., 2000), as in (Brault et al., 2014). We here propose to use the ICL criterion of a model without HMRF. For a model $\mathcal{M}_{gm}$ with g row clusters and m column clusters, we compute ICL with (5), using $\widetilde{\boldsymbol{Z}}$ and $\widetilde{\boldsymbol{W}}$, the matrices of variational posterior probabilities obtained with VEM.

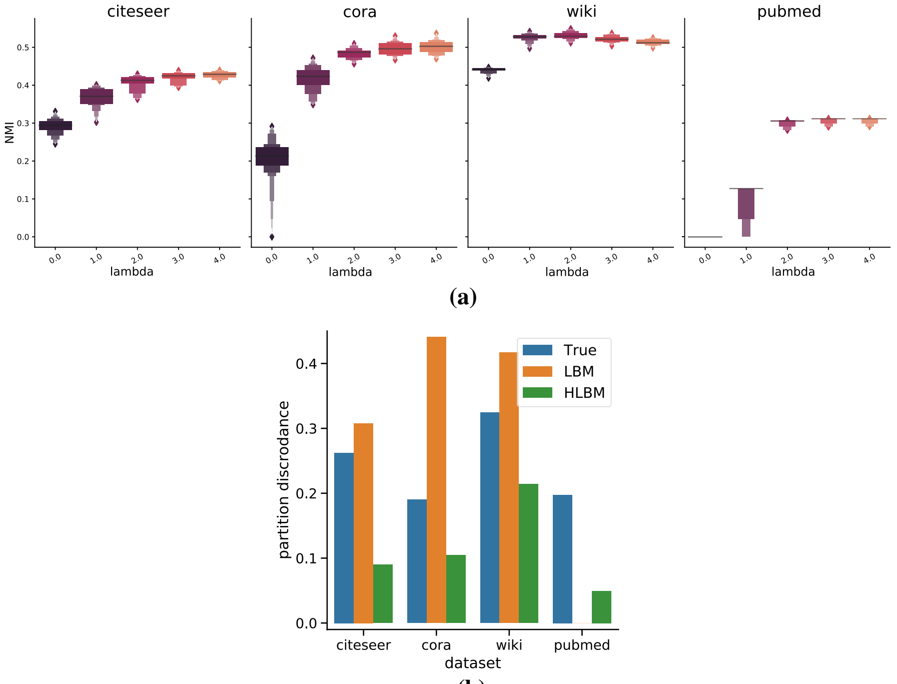
$$\begin{aligned} &\text{ICL}(g,m) = \log p(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{W} | \mathcal{M}_{gm}) \\ &\approx \max_{\boldsymbol{\Theta}} \log p(\boldsymbol{X}, \widetilde{\boldsymbol{Z}}, \widetilde{\boldsymbol{W}} | \boldsymbol{\Theta}, \mathcal{M}_{gm}) - \frac{g-1}{2} \log n - \frac{m-1}{2} \log d - \frac{gm}{2} \log(nd). \end{aligned} \tag{5}$$

We computed the ICL for each dataset with $g \in \{g_{\text{True}} - 4, \dots, g_{\text{True}} + 4\}$ and $m \in \{4, \dots, 12\}$ to determine both g and m. The results are presented in Appendix 1. Since the clustering problems are difficult, and even if the algorithm compares well to the literature, some classes that the algorithm can not distinguish are merged. This results in an underestimated number of row cluster.

In the following experiment (see Sect. 6.4), we set the number of row clusters to its true value, and determine an appropriate number of column clusters m_{ICL} using the ICL. We found respectively 6, 7, 4 and 5 column clusters for Cora, Citeseer, Wiki and Pubmed with the ICL criterion.

6.3 Setting the hyper-parameters

For all the datasets, we use our VEM algorithm with $\eta = 0.7$. Based on our study on simulated data, we set $\lambda_r = 3$ and use a symmetric adjacency matrix $\boldsymbol{A}$. As can be seen in Fig. 5a, the algorithm benefits from a positive value of λ in terms of clustering performance, and the choice of λ_r does not appear to be too critical in the range [1, 4], except for Pubmed which seems to require more regularization. In fact, we observe that our algorithm does not recover any meaningful structure with $\lambda_r = 0$ on Pubmed, probably due to a low



(a)

(b)

Fig. 5 **a** Row clustering performances in terms of NMI for different values of λ_r on the datasets. **b** Discordance criteria $\mathcal{R}(\boldsymbol{A}, \mathcal{P})$ with $\mathcal{P}$ denoting, for each dataset: the true partition (True), the partition obtained with $\lambda_r = 0$ (LBM) and the partition obtained with $\lambda_r = 3$ (HLBM)

class separability related to the relatively low dimension of the word vectors, but the addition of the adjacency matrix allows it to overcome this issue.

6.4 Evaluation of the CEM and VEM algorithms in terms of clustering

We compare our algorithms (Poisson HLBM VEM: `PHLBMVEM` and Poisson HLBM CEM: `PHLBMCEM`) to the following *deep learning* algorithms: `GAE` (Kipf & Welling, 2016b), `VGAE` (Kipf & Welling, 2016a), `MGAE` (Wang et al., 2017), `ARGA` and `ARVGA` (Pan et al., 2018), `AGC` (Zhang et al., 2019) and `DAEGC` (Wang et al., 2019). All these algorithms are unsupervised, which enables a fair comparison. Further comparisons to weakly-supervised methods are presented in Appendix 1. As discussed in Sect. 2 and Appendix 1, the `CITTC` model of Song et al. (2010) is a special case of ours. Consequently, we do not compare our algorithms to `CITTC`. We run our algorithms 20 times and report.[3] Otherwise, results are reported from the original paper the results in terms of clustering accuracy (ACC) and Normalized Mutual Information (NMI) on Table 2.

We observe that our algorithms perform well compared with most of these more complex algorithms. We also note that CEM performs comparably to VEM on this task. In

[3] Algorithms or results marked by * are reported from (Zhang et al., 2019).

Table 2 Attributed network clustering metrics (mean std, higher is better)

Method	Cora		Citeseer		Wiki		Pubmed	
	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
GAE(*)	53.25	40.69	41.26	18.34	17.33	11.93	64.08	22.97
VGAE(*)	55.95	38.85	44.38	22.71	28.67	30.28	65.48	25.09
ARGE	64.00	44.90	57.3	35.0	41.40(*)	39.50(*)	59.12(*)	23.17(*)
ARVGE	63.80	45.00	54.4	26.10	41.55(*)	40.01(*)	58.22(*)	20.62(*)
MGAE	63.43	45.57	63.56	39.75	50.14	47.97	43.88(*)	8.60(*)
DAEGC	**70.04**	52.8	67.20	39.70	N/A	N/A	67.10	26.60
AGC	68.92	**53.68**	67.00	41.13	47.65	45.28	**69.78**	**31.59**
PHLBMCEM	68.6 ± 1.9	49.8 ± 0.9	66.2 ± 1.9	40.8 ± 1.4	53.3 ± 3.7	51.9 ± 1.3	67.4 ± 0.6	30.8 ± 0.9
PHLBMVEM	65.9 ± 2.6	49.7 ± 1.6	**67.6 ± 1.2**	**42.1 ± 1.3**	**54.8 ± 2.4**	**52.2 ± 0.9**	67.0 ± 1.6	30.9 ± 0.8

Results in bold are correspond to the best performing method

comparison to the baseline deep-learning models, ours has the advantage of being interpretable, parsimonious, to rely on a simple input parameter (λ_r) and to be able to perform model selection.

We can measure the information conveyed by the adjacency matrix, considered as a ML constraints matrix. In Fig. 5b, we observe, using the true clusters as reference, that approximately 70% of the edges of the adjacency matrices actually encode a true ML relationship. We also find that an important fraction of the ML relationships of $\boldsymbol{A}$ are not already inferred by the algorithm with $\lambda_r = 0$ in the data (the LBM returns only one non-empty cluster on Pubmed, so all ML relationships are satisfied). Finally, we note that most of the ML constraints are satisfied after fitting the model with $\lambda_r = 3$. The results are more nuanced on Wiki, where $\boldsymbol{A}$ encodes less true ML relationships and the partition learned with the HLBM only satisfies approximately 80% of the constraints of $\boldsymbol{A}$. This is probably due to the high true number of row clusters $g = 17$, that makes a random ML relationship less likely to be true.

7 Conclusion

We have introduced a general probabilistic framework for co-clustering that incorporates ML and CL relationships in the LBM based on HMRF. We presented two efficient inference algorithms based on Variational and Classification EM that also benefit from the supervision in the initialization. We showed that we can establish connections between our algorithms and GCNs as well as manifold regularization. We instantiated this framework on a model for count data and presented detailed VEM and CEM algorithms for which we analyzed the time and space complexity. We have studied the behavior of these algorithms on simulated data when varying the tradeoff between the discriminative and the generative component of the model. Our algorithms have also demonstrated good clustering performances in comparison with *deep learning* algorithms devoted to the task of attributed network clustering.

In future work, the model can be extended to represent more complex relationships in the latent space with a more general MRF. Stochastic variants of the EM algorithm based on simulated field EM (Celeux et al., 2003) could also be investigated. Finally, the definition of column constraints in the context of clustering with pairwise semi-supervision, only available in the row space, is an important problem-specific research track.

Appendix

1. The partition function

We here compute the partition function $\Gamma_r(\Xi^r)$. Let $\mathcal{Z}$ be the set of partitions of n rows in g clusters. $\Gamma_r(\Xi^r)$ must ensure that $\sum_{\boldsymbol{Z}\in\mathcal{Z}} p(\boldsymbol{Z};\boldsymbol{\Theta}) = 1$. Let $\mathcal{Z}_{Y_r}$ be the set of row partitions in g clusters for the rows for which we have prior knowledge and let $\mathcal{Z}_{\overline{Y}_r}$ be the set of row partitions in g clusters for the rows for which we do not. Note that $\mathcal{Z}$ can be decomposed as $\mathcal{Z}_{Y_r} \times \mathcal{Z}_{\overline{Y}_r}$. Using the definition of the potentials in (1), we have:

$$\begin{aligned}
\Gamma_r &= \sum_{\boldsymbol{Z}\in\mathcal{Z}} \exp\Big(\sum_{i\notin Y_r} \log \alpha_{z_i} + \frac{1}{2}\sum_{i\in Y_r}\sum_{i'\in\mathcal{N}_r(i)} \log \psi^r_{ii'}(z_i, z_{i'};\Xi^r)\Big) \\
&= \Big(\sum_{\boldsymbol{Z}'\in\mathcal{Z}_{\overline{Y}_r}} \prod_{i\notin Y_r} \alpha_{z'_i}\Big) \times \Big(\sum_{\boldsymbol{Z}''\in\mathcal{Z}_{Y_r}} \exp\Big(\frac{1}{2}\sum_{i\in Y_r}\sum_{i'\in\mathcal{N}_r(i)} \log \psi^r_{ii'}(z''_i, z''_{i'};\Xi^r)\Big)\Big) \\
&= \sum_{\boldsymbol{Z}\in\mathcal{Z}_{Y_r}} \exp\Big(\frac{1}{2}\sum_{i\in Y_r}\sum_{i'\in\mathcal{N}_r(i)} \log \psi^r_{ii'}(z_i, z_{i'};\Xi^r)\Big),
\end{aligned}$$

since $\sum_{\boldsymbol{Z}'\in\mathcal{Z}_{\overline{Y}_r}} \prod_{i\notin Y_r} \alpha_{z'_i} = 1$; as it is the sum of the density of $n - |Y_r|$ independent categorical distributions on its support. Thereby, Γ_r only depends on Ξ^r.

2. Connection to CITTC (Song et al., 2010)

The model of `CITTC` is based on `ITTC` with a Kullback-Leibler divergence, that consists in maximizing the KL divergence between the empirical distribution on the set of word×documents and a distribution (on this same set) that factorizes using the latent variables. It has been shown in (Govaert & Nadif, 2018) that this model is equivalent to the LBM with equal mixture proportions and a Poisson mixture distribution.

Regarding the HMRF regularization of `CITTC`, the weight matrices of the potentials are defined as:

$$\xi^r_{ii'} = \begin{cases} D_{KL}(p_i^{\mathrm{emp}} || p_{i'}^{\mathrm{emp}}) & (i,i') \in \mathcal{M}^r \\ \big(D_{\max} - D_{KL}(p_i^{\mathrm{emp}} || p_{i'}^{\mathrm{emp}})\big) & (i,i') \in \mathcal{C}^r \\ 0 & \text{otherwise} \end{cases}$$

where p_i^{emp} is the empirical multinomial distribution over the column space, with $p_i^{\mathrm{emp}}(j) = \frac{x_{ij}}{x_{i.}}$, and $D_{\max} = \max_{(i,i')} D_{KL}(p_i^{\mathrm{emp}} || p_{i'}^{\mathrm{emp}})$. The authors also propose to set $\lambda_r = n^{-\frac{1}{2}}$ and $\lambda_c = d^{-\frac{1}{2}}$. It could thus be seen as a special case of our model.

However, this approach has several limitations. First, the Kullback-Leibler divergence is not symmetric, which leads to unsymmetrical potentials, requiring less convenient computations and making two different values for the penalty for the dissatisfaction of a given constraint. Second, the computation of the proposed potentials is not straightforward for sparse matrices. Finally, the model using these potentials is not generative anymore since sampling the latent variables now depends on the data matrix.

3. Objective function of the VEM algorithm

The objective function of the VEM algorithm given by $F(\widetilde{\mathbf{Z}}, \widetilde{\mathbf{W}}, \boldsymbol{\Theta})$ is a lower-bound of the log-likelihood of the model $\ell(\boldsymbol{\Theta})$:

$$\begin{aligned}
F(\widetilde{\mathbf{Z}}, \widetilde{\mathbf{W}}, \boldsymbol{\Theta}) &= \mathbb{E}_Q\big(\log p(\mathbf{X}, \mathbf{Z}, \mathbf{W}; \boldsymbol{\Theta})\big) + H(Q) \\
&= \mathbb{E}_Q\Big(\log \frac{p(\mathbf{X}, \mathbf{Z}, \mathbf{W}; \boldsymbol{\Theta})}{Q(\mathbf{Z}, \mathbf{W}; \widetilde{\mathbf{Z}}, \widetilde{\mathbf{W}})}\Big) \\
&\leq \log \mathbb{E}_Q\Big(\frac{p(\mathbf{X}, \mathbf{Z}, \mathbf{W}; \boldsymbol{\Theta})}{Q(\mathbf{Z}, \mathbf{W}; \widetilde{\mathbf{Z}}, \widetilde{\mathbf{W}})}\Big) \\
&= \log \sum_{\mathbf{Z}, \mathbf{W} \in \mathcal{Z} \times \mathcal{W}} Q(\mathbf{Z}, \mathbf{W}; \widetilde{\mathbf{Z}}, \widetilde{\mathbf{W}}) \frac{p(\mathbf{X}, \mathbf{Z}, \mathbf{W}; \boldsymbol{\Theta})}{Q(\mathbf{Z}, \mathbf{W}; \widetilde{\mathbf{Z}}, \widetilde{\mathbf{W}})} \\
&= \ell(\boldsymbol{\Theta})
\end{aligned}$$

where the third line is obtained using Jensen's inequality. We first note that $\mathbb{E}_Q(z_{ik}) = \tilde{z}_{jk}$. Then, since $Q(\mathbf{Z}, \mathbf{W}; \widetilde{\mathbf{Z}}, \widetilde{\mathbf{W}}) = Q(\mathbf{Z}; \widetilde{\mathbf{Z}}) Q(\mathbf{W}; \widetilde{\mathbf{W}})$, we have $H(Q) = H(\widetilde{\mathbf{Z}}) + H(\widetilde{\mathbf{W}})$. Finally, as the variational distribution considers independent latent variables, we have $H(\widetilde{\mathbf{Z}}) = -\sum_{ik} \tilde{z}_{ik} \log \tilde{z}_{ik}$. Thus, the objective function of the VEM algorithm takes the following form:

$$\begin{aligned}
F(\widetilde{\mathbf{Z}}, \widetilde{\mathbf{W}}, \boldsymbol{\Theta}) = &\sum_{i \notin Y_r} \sum_k \tilde{z}_{ik} \log \alpha_k + \frac{\lambda_r}{2} \sum_{ii'} s^r_{ii'} \sum_k \tilde{z}_{ik} \tilde{z}_{i'k} - \sum_{ik} \tilde{z}_{ik} \log \tilde{z}_{ik} \\
&+ \sum_{j \notin Y_c} \sum_{\ell} \tilde{w}_{j\ell} \log \beta_\ell + \frac{\lambda_c}{2} \sum_{jj'} s^c_{jj'} \sum_{\ell} \tilde{w}_{j\ell} \tilde{w}_{j'\ell} - \sum_{j\ell} \tilde{w}_{j\ell} \log \tilde{w}_{j\ell} \\
&+ \sum_{ijk\ell} \tilde{z}_{ik} \tilde{w}_{j\ell} \log \phi(x_{ij}; \epsilon_{ijk\ell}).
\end{aligned}$$

4. Model selection

The experiment on the ICL criterion for model selection are presented in Fig. 6. As an illustration of the underestimation of the number of row clusters with the ICL, the clustering row confusion matrices obtained for Cora are presented below.

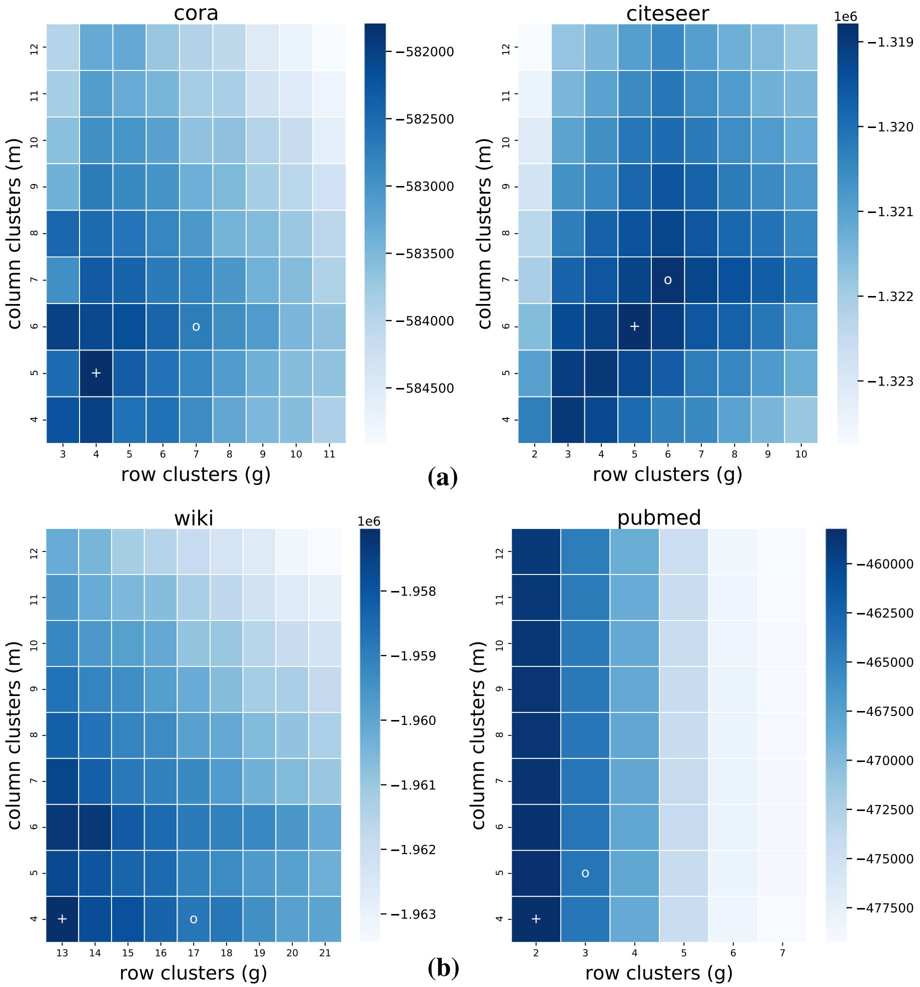


Fig. 6 Maximal ICL obtained on Cora, Citeseer, Wiki and Pubmed as a function of the number of row and column clusters. Cells marked with "+" correspond to the (g, m) with the greatest ICL. Cells marked with "o" correspond to the m with the greatest ICL with g set to its true value

$$C_{g_{\mathrm{True}}} = \begin{pmatrix} 506 & 58 & 72 & 20 & 137 & 19 & 6 \\ 16 & 150 & 146 & 10 & 24 & 1 & 4 \\ 0 & 0 & 142 & 38 & 0 & 0 & 0 \\ 7 & 1 & 13 & 183 & 3 & 0 & 10 \\ 18 & 10 & 44 & 6 & 310 & 38 & 0 \\ 0 & 0 & 95 & 12 & 4 & 181 & 6 \\ 15 & 0 & 11 & 20 & 4 & 1 & 367 \end{pmatrix}, C_{g_{\mathrm{ICL}}} = \begin{pmatrix} 560 & 120 & 0 & 0 & 131 & 0 & 7 \\ 12 & 317 & 0 & 0 & 12 & 0 & 10 \\ 0 & 142 & 0 & 0 & 38 & 0 & 0 \\ 8 & 197 & 0 & 0 & 4 & 0 & 8 \\ 30 & 48 & 0 & 0 & 348 & 0 & 0 \\ 0 & 145 & 0 & 0 & 146 & 0 & 7 \\ 3 & 36 & 0 & 0 & 6 & 0 & 373 \end{pmatrix},$$

Here, $C_{g_{\mathrm{True}}}$ is obtained using the true number of row clusters and $C_{g_{\mathrm{ICL}}}$ is obtained using the number of row clusters found with the ICL. For both matrices, $C_{kk'}$ is the number of

Table 3 Attributed network clustering accuracies (ACC)(mean ± std, higher is better) in a transductive setup

	Cora	Citeseer	Pubmed
LP	68.0	45.3	63.0
PLANETOID	75.7	64.7	77.2
DEEPWALK	70.7 ± 0.6	51.4 ± 0.5	74.3 ± 0.9
DGI	82.3 ± 0.6	71.8 ± 0.7	76.8 ± 0.6
PHLBMCEM	68.6 ± 1.9	66.2 ± 1.9	67.4 ± 0.6
PHLBMVEM	65.9 ± 2.6	67.6 ± 1.2	67.0 ± 1.6

The results are reported from (Yang et al., 2016) and (Veličković et al., 2019) points known to be in the true cluster k and predicted to be in cluster k' of the algorithm. We observe on $C_{g_{\mathrm{ICL}}}$ that the true cluster of the 2nd and 5th rows are better recovered for $g = g_{\mathrm{ICL}}$ than for $g = g_{\mathrm{True}}$. We also note on $C_{g_{\mathrm{ICL}}}$ that the true clusters of the rows 3, 4 and 6 are merged into the algorithm clusters of the columns 2 and 5. Thus, the algorithm with $g = g_{\mathrm{True}}$ correctly recovers certain clusters but merges the others, while with $g = g_{\mathrm{ICL}}$ these clusters are not so well recovered but no cluster is merged. This explains why the ICL criterion leads to an underestimated number of row clusters.

5. Comparison to supervised approaches

We here provide a comparison of our approach to some supervised algorithms. First, some ANE algorithms such as deep graph infomax (DGI) (Veličković et al., 2019) and (DEEPWALK) (Perozzi et al., 2014) learn a representation of the nodes in an unsupervised way but evaluate their classification performances using a supervised learning algorithm on the learned representation. Besides, we add comparison to weekly-supervised algorithms such as Label Propagation (LP) (Zhu et al., 2003) and PLANETOID (Yang et al., 2016) (Table 3).

Acknowledgements We thank the three anonymous reviewers for their detailed comments that have helped us a lot to improve this manuscript.

Author contributions All authors contributed to the study conception and design. Material preparation and data analysis were performed by Paul Riverain. The first draft of the manuscript was written by Paul Riverain and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding This work was partly funded by the CIFRE program of the French Ministry of Higher Education and Research.

Data availability The data is publicly available online.

Code availability The code is not made available.

Declarations

Conflict of interest There is no conflict of interest.

Ethical approval Not Applicable.

Consent to participate Not Applicable.

Consent for publication All authors consent to publish this manuscript.

References

Affeldt, S., Labiod, L., & Nadif, M. (2021). Regularized bi-directional co-clustering. *Statistics and Computing, 31*(3), 32.

Ailem, M., Role, F., & Nadif, M. (2017). Model-based co-clustering for the effective handling of sparse data. *Pattern Recognition, 72*, 108–122.

Ambroise, C., & Govaert, G. (1998). Convergence of an em-type algorithm for spatial clustering. *Pattern Recognition Letters, 19*(10), 919–927.

Banerjee, A., Dhillon, I., Ghosh, J., Merugu, S., & Modha, D. S. (2004). A generalized maximum entropy approach to bregman co-clustering and matrix approximation. In: KDD (p. 509)

Basu, S., Bilenko, M., & Mooney, R. J. (2004). A probabilistic framework for semi-supervised clustering. In *SIGKDD* (pp. 59–68)

Basu, S., Davidson, I., & Wagstaff, K. (2008). *Constrained clustering: Advances in algorithms, theory, and applications* (1st ed.). Chapman and Hall.

Besag, J. (1986). On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society: Series B (Methodological), 48*(3), 259–279.

Biernacki, C., Celeux, G., & Govaert, G. (2000). Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 22*(7), 719–725.

Bilenko, M., & Basu, S. (2004). A comparison of inference techniques for semi-supervised clustering with hidden markov random fields. In *ICML-2004 Workshop on Statistical Relational Learning and its Connections to Other Fields (SRL-2004).*

Bock, H. H. (2020). Co-clustering for object by variable data matrices. In T. Imaizumi & A. Nakayama (Eds.), *Advanced studies in behavior metrics and data science* (pp. 3–17). Springer.

Brault, V., Keribin, C., Celeux, G., & Govaert, G. (2014). Estimation and selection for the latent block model on categorical data. *Statistics and Computing, 25*, 1–16.

Celeux, G., & Govaert, G. (1992). A classification EM algorithm for clustering and two stochastic versions. *Computational Statistics and Data Analysis, 14*(3), 315–332.

Celeux, G., Forbes, F., & Peyrard, N. (2002). EM-based image segmentation using Potts models with external field. Research Report RR-4456, INRIA

Celeux, G., Forbes, F., & Peyrard, N. (2003). Em procedures using mean field-like approximations for markov model-based image segmentation. *Pattern Recognition, 36*(1), 131–144.

Davidson, I., Wagstaff, K. L., & Basu, S. (2006). Measuring constraint-set utility for partitional clustering algorithms. In J. Fürnkranz, T. Scheffer, & M. Spiliopoulou (Eds.), *Knowledge discovery in databases: PKDD* (pp. 115–126). Springer.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society Series B (Methodological), 39*(1), 1–38.

Deodhar, M., & Ghosh, J. (2010). Scoal: A framework for simultaneous co-clustering and learning from complex data. *ACM Transactions on Knowledge Discovery from Data, 4*(3), 1–31.

Dhillon, IS., Mallela, S., & Modha, DS. (2003). Information-theoretic co-clustering. In *SIGKDD* (pp. 89–98).

Govaert, G., & Nadif, M. (2005). An EM algorithm for the block mixture model. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 27*(4), 643–647.

Govaert, G., & Nadif, M. (2008). Block clustering with bernoulli mixture models: Comparison of different approaches. *Computational Statistics & Data Analysis, 52*(6), 3233–3245.

Govaert, G., & Nadif, M. (2013). *Co-clustering: Models, algorithms and applications*. Wiley.

Govaert, G., & Nadif, M. (2018). Mutual information, phi-squared and model-based co-clustering for contingency tables. *Advances in Data Analysis and Classification, 12*(3), 455–488.

He, X., Cai, D., Shao, Y., Bao, H., & Han, J. (2011). Laplacian regularized gaussian mixture model for data clustering. *IEEE Transactions on Knowledge and Data Engineering, 23*(9), 1406–1418.

Hinton, GE., Osindero, S., Bao, K. (2005). Learning causally linked markov random fields. In *The 10th International Workshop on AISTATS* (pp. 128–135).

Hubert, L., & Arabie, P. (1985). Comparing partitions. *Journal of Classification, 2*, 193.

Kilic, K., Tanaka, D., Honda, K., Ubukata, S., & Notsu, A. (2016). A semi-supervised framework for mmms-induced fuzzy co-clustering with virtual samples. *Advances in Fuzzy Systems, 2016*, 5206048.

Kipf, TN., & Welling, M. (2016a). Semi-supervised classification with graph convolutional networks. http://arxiv.org/1609.02907

Kipf, TN., & Welling, M. (2016b). Variational graph auto-encoders. In *NIPS Workshop on Bayesian Deep Learning*.

Lange, T., Law, MHC., Jain, AK., & Buhmann, JM. (2005). Learning with constrained and unlabelled data. In *CVPR'05 1* (Vol. 1, pp. 731–738).

Madeira, S., & Oliveira, A. (2004). Biclustering algorithms for biological data analysis: A survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics, 1*, 24–45.

Miele, V., Picard, F., & Dray, S. (2014). Spatially constrained clustering of ecological networks. *Methods in Ecology and Evolution, 5*(8), 771–779.

Nam, J. H., Couch, D., da Silveira, W. A., Yu, Z., & Chung, D. (2020). Palmer: Improving pathway annotation based on the biomedical literature mining with a constrained latent block model. *BMC Bioinformatics, 21*(1), 432.

Pan, S., Hu, R., Long, G., Jiang, J., Yao, L., & Zhang, C. (2018). Adversarially regularized graph autoencoder for graph embedding. In *IJCAI, International Joint Conferences on Artificial Intelligence Organization* (pp. 2609–2615).

Pensa, R. G., & Boulicaut, J. F. (2008). Constrained co-clustering of gene expression data. In *SIAM International Conference on Data Mining* (pp. 25–36).

Perozzi, B., Al-Rfou, R., & Skiena, S. (2014). Deepwalk: Online learning of social representations. In: *SIGKDD* (pp. 701–710).

Robert, V., Vasseur, Y., & Brault, V. (2021). Comparing high-dimensional partitions with the co-clustering adjusted rand index. *Journal of Classification, 38*(1), 158–186.

Salah, A., & Nadif, M. (2017). Social regularized von mises–Fisher mixture model for item recommendation. *Data Mining and Knowledge Discovery, 31*(5), 1218–1241.

Salah, A., Ailem, M., & Nadif, M. (2018). Word co-occurrence regularized non-negative matrix tri-factorization for text data co-clustering. In *AAAI* (pp. 3992–3999).

Song, Y., Pan, S., Liu, S., Wei, F., Zhou, M., & Qian, W. (2010). Constrained coclustering for textual documents. In *AAAI* (Vol. 24, No. 1).

Tang, W., Lu, Z., & Dhillon, IS. (2009). Clustering with multiple graphs. In *ICDM* (pp. 1016–1021).

Van Mechelen, I., Bock, H. H., & De Boeck, P. (2004). Two-mode clustering methods: A structured overview. *Statistical Methods in Medical Research, 13*(5), 363–394.

Veličković, P., Fedus, W., Hamilton, WL., Liò, P., Bengio, Y., & Hjelm, RD. (2019). Deep graph infomax. In *International Conference on Learning Representations*.

Wang, C., Pan, S., Long, G., Zhu, X., & Jiang, J. (2017). Mgae: Marginalized graph autoencoder for graph clustering. In*CIKM '17* (pp. 889–898).

Wagstaff, K., Cardie, C., Rogers, S., & Schrödl, S. (2001). Constrained k-means clustering with background knowledge. In *ICML* (pp. 577–584).

Wang, C., Pan, S., Hu, R., Long, G., Jiang, J., & Zhang, C. (2019). Attributed graph clustering: A deep attentional embedding approach. http://arxiv.org/1906.06532

Yan, Y., Chen, L., & Tjhi, W. C. (2013). Fuzzy semi-supervised co-clustering for text documents. *Fuzzy Sets and Systems, 215*, 74–89.

Yang, Z., Cohen, WW., & Salakhutdinov, R. (2016). Revisiting semi-supervised learning with graph embeddings. In *ICML* (pp. 40–48).

Yu, X., Yu, G., Wang, J., & Domeniconi, C. (2019). Co-clustering ensembles based on multiple relevance measures. In *IEEE Transactions on Knowledge and Data Engineering*.

Zhang, X., Liu, H., Li, Q., & Wu, XM. (2019). Attributed graph clustering via adaptive graph convolution. In *IJCAI-19* (pp. 4327–4333).

Zhu, X., & Lafferty, J. (2005). Harmonic mixtures: Combining mixture models and graph-based methods for inductive and scalable semi-supervised learning. In *ICML* (pp. 1052–1059).

Zhu, X., Ghahramani, Z., & Lafferty, J. (2003). Semi-supervised learning using gaussian fields and harmonic functions. In *ICML* (pp. 912–919).

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.