Data Mining and Knowledge Discovery
https://doi.org/10.1007/s10618-021-00787-z

BROCCOLI: overlapping and outlier-robust biclustering through proximal stochastic gradient descent

Sibylle Hess[1] · Gianvito Pio[2,3] · Michiel Hochstenbach[1] · Michelangelo Ceci[2,3,4]

Received: 22 November 2020 / Accepted: 21 July 2021

Abstract

Matrix tri-factorization subject to binary constraints is a versatile and powerful framework for the simultaneous clustering of observations and features, also known as biclustering. Applications for biclustering encompass the clustering of high-dimensional data and explorative data mining, where the selection of the most important features is relevant. Unfortunately, due to the lack of suitable methods for the optimization subject to binary constraints, the powerful framework of biclustering is typically constrained to clusterings which partition the set of observations or features. As a result, overlap between clusters cannot be modelled and every item, even outliers in the data, have to be assigned to exactly one cluster. In this paper we propose BROCCOLI, an optimization scheme for matrix factorization subject to binary constraints, which is based on the theoretically well-founded optimization scheme of proximal stochastic gradient descent. Thereby, we do not impose any restrictions on the obtained clusters. Our experimental evaluation, performed on both synthetic and real-world data, and against 6 competitor algorithms, show reliable and competitive

Responsible editor: Annalisa Appice, Sergio Escalera, Jose A. Gamez, Heike Trautman

✉ Gianvito Pio
gianvito.pio@uniba.it

Sibylle Hess
s.c.hess@tue.nl

Michiel Hochstenbach
m.e.hochstenbach@tue.nl

Michelangelo Ceci
michelangelo.ceci@uniba.it

[1] Department of Mathematics and Computer Science, TU Eindhoven, 5600 MB Eindhoven, The Netherlands

[2] Department of Computer Science, University of Bari Aldo Moro, Via Orabona, 4, Bari, Italy

[3] Big Data Lab, National Interuniversity Consortium for Informatics (CINI), Rome, Italy

[4] Jozef Stefan Institute, Jamova 39, Ljubljana, Slovenia

Published online: 11 August 2021

performance, even in presence of a high amount of noise in the data. Moreover, a qualitative analysis of the identified clusters shows that Broccoli may provide meaningful and interpretable clustering structures.

Keywords Biclustering · Co-clustering · Proximal stochastic gradient descent · Matrix tri-factorization

1 Introduction

In the field of clustering, and more generally in data mining, one of the biggest open problems is the optimization subject to binary constraints. The imposition of binary constraints in the objective of the optimization corresponds, in the context of clustering, to requiring that each observation clearly belongs to one cluster or not, with no gray areas. More generally, binary constraints arise from a request for interpretable and definite data mining results: is a picture showing a cat? Should a movie be recommended to this user? Should the next chess move be this one? Binary results provide definite yes or no answers to the questions arising when solving data mining tasks.

Many methods are able to solve binary-constrained problems. However, they mostly work under one condition: *exclusivity*. In particular, we are assuming that if a picture shows a cat, then it cannot show a dog; if a movie is assigned to one cluster (e.g., a genre), then it cannot belong to another cluster (i.e., to another genre); there should be only one possible next chess move, which is the optimal one. From these examples, we can easily observe that the exclusivity assumption may make sense or not depending on the specific application. For example, for the movie genre case, and for movie recommendation in general, the exclusivity assumption is inappropriate: there is no one-to-one relation between genres and groups of movies, or between groups of users and movies. This scenario provides an ideal motivation for the fundamental contribution of the *biclustering* task: the simultaneous clustering (identification of groups) of users and movies, where a *bicluster* is a selection of users and movies, such that the users give similar ratings for the movies in the group, and the movies have similar ratings from the users in the group. Unfortunately, this motivation only works for the philosophy *behind* biclustering; many *algorithms for* biclustering impose the exclusivity constraint, even though it is no necessarily required by the task definition. Provided that they are not constrained by such an exclusivity assumption, biclusters could, e.g., represent a group of science-fiction fans and science-fiction movies. A science-fiction fan (usually) does not exclusively like science-fiction movies. Analogously, a science-fiction movie is not exclusively watched by science-fiction fans. In this respect, the exclusivity assumption is clearly imposing stringent, unrealistic, constraints.

Similar observations can be drawn in other biclustering applications. For example, biclustering of gene-expression data is employed to identify groups of genes and patients, which are strongly linked by similar expression levels. Such an analysis can be used to discover functions of genes related to clinical traits. However, one gene generally does not have one single function in an organism, but is actually involved in multiple biological processes (Pio et al. 2015). On the other hand, not every gene necessarily plays a significant role in the considered conditions. In this case, the exclusivity assumption would force every gene to belong to one cluster. Hence, *outliers*, or *isolated objects*, could be improperly modelled in presence of the exclusivity assumption.

A popular way to circumvent the difficulties of binary optimization is to relax the binary constraint into a nonnegative and/or orthogonal one (cf. Sect. 3.2). However, the resulting *fuzzy* clusters are not always easy to interpret and are usually difficult to process automatically.

In this paper we propose the method BROCCOLI (Binary RObust Co-Clustering Optimization through alternating LInearized minimization) to obtain models which can handle cluster overlap and the presence of outliers. BROCCOLI employs a penalization approach, where the relaxed objective is optimized while the violation of binary constraints is penalized. The penalization degree is optimized during the training, such that the returned minimizer of the objective function satisfies the binary constraints. Our optimization method is based on the theoretically founded framework of Proximal Stochastic Gradient Descent (PSGD) for matrix factorizations (Driggs et al. 2020).

We evaluate BROCCOLI on synthetic and real-world data, showing that it is able to detect biclusterings of various structures, being robust to the noise in the data. A qualitative inspection reveals that BROCCOLI is able to derive meaningful clusters, which are interpretable by their modular structure.

2 Background of biclustering models

The biclustering task is relevant when relationships in both observations and features (or rows and columns) have to be captured. Several biclustering approaches have been proposed in the literature, including spectral methods (Dhillon 2001; Kluger 2003), iterative greedy methods (Pio et al. 2012, 2013), matrix factorization methods (Long et al. 2005; Yoo and Choi 2010; Del Buono and Pio 2015), and multi-type methods (Barracchia et al. 2020).

In this paper, we focus on biclustering models based on matrix factorization. Matrix factorization is applied in multiple contexts, including image reconstruction (Zhou and Qi 2011; Yokota et al. 2019) and data representation (Cai et al. 2011; Wang et al. 2018). It is also intrinsically linked to clustering, due to its recognized role as a general framework for clustering objectives (Ding et al. 2006a; Pompili et al. 2014).

Existing clustering algorithms based on matrix factorization mainly rely on an iterative optimization approach, that aims to find a factorization of the input data matrix into two or more matrices that provide cluster membership information. Specifically, given a data matrix $D \in \mathbb{R}^{m\times n}$, there exist suitable numerical methods to optimize a nonnegative matrix tri-factorization:

$$\min_{X,Y,C} \|D - YCX^\top\|^2 \qquad \text{s.t. } X \in \mathbb{R}_+^{n\times r_x},\ Y \in \mathbb{R}_+^{m\times r_y},\ C \in \mathbb{R}_+^{r_y\times r_x},$$

where the matrix Y has an interpretation as a fuzzy row-cluster indicator, X as a fuzzy column-cluster indicator, r_y is the number of row-clusters, and r_x is the number of

Table 1 Overview of biclustering objectives based on matrix factorization

Biclustering	$\min_{X,C,Y} \|D - YCX^\top\|^2$ s.t.
Checkerboard	$X \in \mathbb{1}^{n\times r_x},\ Y \in \mathbb{1}^{m\times r_y},\ C \in \mathbb{R}^{r_y\times r_x}$
Block Diagonal	$X \in \mathbb{1}^{n\times r},\ Y \in \mathbb{1}^{m\times r},\ C = \operatorname{diag}(C_{11},\ldots,C_{rr})$
Binary	$X \in \{0,1\}^{n\times r},\ Y \in \{0,1\}^{m\times r},\ C = I$
Broccoli	$X \in \{0,1\}^{n\times r},\ Y \in \{0,1\}^{m\times r},\ C \in \mathbb{R}^{r\times r}$

column-clusters. The matrix C is the core matrix that assigns a weight to each (fuzzy) bicluster, i.e., to each pair of a row cluster and a column cluster. In particular, Y_{js} generally represents the degree of membership of the jth row of the data matrix to the row cluster s, while X_{it} represents the degree of membership of the column i of the data matrix to the column cluster t.

Depending on the application, different constraints can be imposed. For example, if we force Y to be binary and to have orthogonal columns, then the objective above becomes equivalent to the one of k-means clustering (Bauckhage 2015). In this case, the matrix Y indicates the assigned cluster for each observation, while the matrix $CX^\top$ represents the centroids. We denote the space of $(m \times r_y)$-dimensional cluster indicator matrices, implementing the exclusivity assumption, with $\mathbb{1}^{m\times r_y}$. We have $Y \in \mathbb{1}^{m\times r_y}$ if and only if $Y \in \{0,1\}^{m\times r_y}$ and $|Y_{j\cdot}| = 1$. In essence, $Y \in \mathbb{1}^{n\times r_y}$ if Y is binary and has orthogonal columns.

If we now want to model biclusters, assigning observations as well as features to clusters, without making use of the exclusivity assumption, then we impose binary constraints on the matrices X and Y:

$$\min_{X,Y,C} \|D - YCX^\top\|^2 \qquad \text{s.t. } X \in \{0,1\}^{n\times r_x},\ Y \in \{0,1\}^{m\times r_y},\ C \in \mathbb{R}_+^{r_y\times r_x}.$$

In this case, $Y_{js} = 1$ (resp., $Y_{js} = 0$) means that the jth row of the data matrix is (resp., is not) assigned to the row cluster s, while $X_{it} = 1$ (resp., $X_{it} = 0$) means that the column i of the data matrix is (resp., is not) assigned to the column cluster t. Again, the matrix C assigns a weight to each bicluster, i.e., to each pair of a row cluster and a column cluster.

In Table 1, we show three common biclustering models: checkerboard, diagonal and binary biclustering.

We observe that the objectives of checkerboard and diagonal biclustering impose the exclusivity by constraining the binary factor matrices to the $\mathbb{1}$-space. Since the exclusivity assumption enforces row- and column-clusters not to overlap, for checkerboard and diagonal biclusters there exists a permutation of rows and columns, such that each bicluster appears as a coherent block. This is shown in Fig. 1, where the area of the bicluster combining row-cluster s with column-cluster t is filled with one color, representing the value C_{st}. This clustering concept goes back to Hartigan (1972).

A special case of biclustering arises if the data matrix is binary. In this case, a decomposition into two binary matrices (where the core matrix is equal to the identity

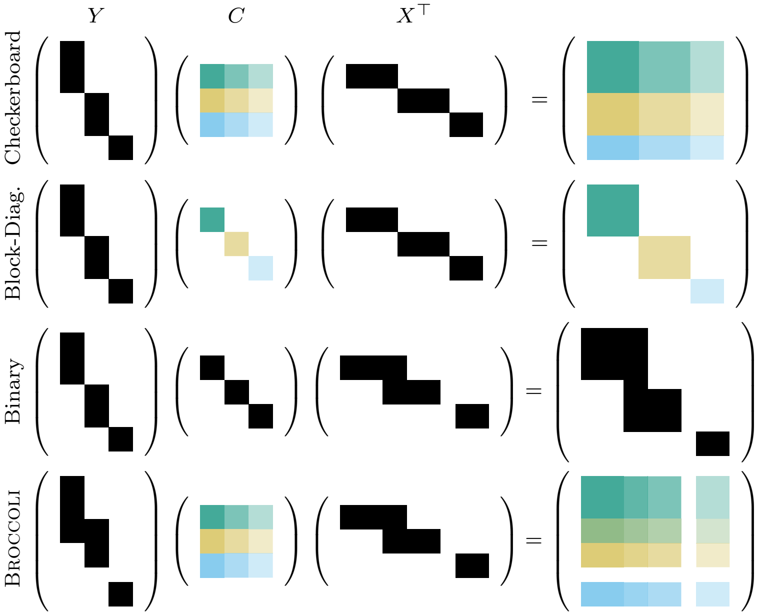

Fig. 1 Visualization of checkerboard (on the top), block-diagonal (second from the top), binary (third from the top, with non-overlapping row-clusters but overlapping column-clusters) and the BROCCOLI (on the bottom) biclustering models. Every bicluster is a combination of a row-cluster and a column-cluster, which we visualize here as a block. Real values are indicated by colors, while binary values are indicated in black and white (black represents a one; white represents a zero). Best viewed in color

matrix) may benefit the interpretability of the result (see Table 1). We refer to such factorizations as binary biclustering. For this task, the exclusivity constraint is usually not imposed on both binary factor matrices. This is because it would lead to a binary block-diagonal model, which is too simplistic for most applications. However, binary biclusters do have an inbuilt penalization of overlap, because overlapping areas of binary biclusters are approximated with a value of 2—introducing an approximation error of at least one. As a result, every pair of overlapping biclusters adds to the approximation error a value equal to the size of the overlapping area. Consequently, binary biclusters have usually either non-overlapping row- or non-overlapping column-clusters, as depicted in Fig. 1 (third row).

In this paper, we aim to get rid of restrictive constraints on the clustering structure. Our objective (see Table 1) resembles that of checkerboard biclustering, with the difference that our cluster indicator matrices do not enforce the exclusivity constraint. The result may yield overlapping row- and column-clusters (as shown in Fig. 1), and identify outlier observations (rows of the data matrix which are not assigned to any row-cluster) and outlier features (columns of the data matrix which are not assigned to any column-cluster).

3 Related work

The major challenge faced by clustering methods, without assuming exclusivity, comes from the optimization phase. While the fuzzy real-valued counterparts can suitably be optimized by numerical methods, the discrete nature of binary clustering tasks does not allow for a direct adaptation of such optimization methods. Deriving definite cluster assignments is an inherently combinatorial complex problem. The major advantage of using the exclusivity assumption is the possibility to adopt an efficient iterative optimization scheme, based on *alternating minimization* (cf. Sect. 3.1). If the exclusivity assumption is supposed to be relaxed, then the usual way is to relax the binary clustering constraints as well, such that numerical optimization methods can be applied (cf. Sect. 3.2). It is noteworthy that also the approach proposed in this paper, for the optimization of non-exclusive clusterings, is based on a relaxation of the constraints. In particular, our approach is related to the penalization approach reviewed in Sect. 3.3. However, in our method, the relaxations are *gradually reversed*, being a part of the optimization, such that we obtain binary solutions in the end.

3.1 Methods based on alternating minimization

Alternating minimization for clustering has been introduced by Lloyd (1982) with the k-means algorithm. Iteratively, one of the factor matrices is optimized while the other factor matrices are kept fixed. The exclusivity assumption enables the analytical derivation of the optimizer in every iteration (Gaul and Schader 1996; Mirkin et al. 1995). In other words, it is not necessary to apply gradient descent at every optimization step, but it is possible to directly state the optimum for one of the matrices. This way, the optimization subject to binary constraints is facilitated. The alternating minimization scheme has been implemented for checkerboard biclustering (Vichi 2001; Wang et al. 2011; Cho et al. 2004) and for diagonal biclustering (Han et al. 2017; Song et al. 2020). Koyutürk and Grama (2003), Li (2005) proposed alternating minimization for binary matrix factorization, where one the factor matrices is constrained to satisfy the exclusivity assumption. In this scenario, row-clusters are always nonoverlapping, but column-clusters may overlap, or vice versa. Although alternating minimization is an elegant and theoretically founded optimization method, it has the drawback that the global, yet separate, minimization in every step tends to converge to a minimum which is not far from the starting point. This behavior makes alternating minimization very sensitive to the initialization (Zhou et al. 2015). In contrast, gradient-based methods are generally more prone to explore the hypothesis space, before converging to a stationary point (assuming that the chosen step-size is not particularly small).

3.2 Methods based on (soft-) orthogonal relaxations

Since the binary constraints in clustering objectives hinder the application of numerical optimization methods, relaxations of the binary constraints have been adopted in the literature. The most popular approaches are based on nonnegative (soft-) orthogonal relaxations (Ding et al. 2006b; Yoo and Choi 2010; Zha et al. 2001; Dhillon 2001; Nie

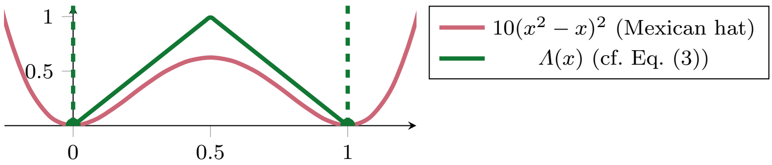

Fig. 2 Binary penalization functions: the Mexican hat function and Λ

et al. 2017). These methods aim at solving the following objective:

$$\min_{Y\geq 0, C\geq 0, X\geq 0} \left\| D - YCX^\top \right\|^2, \text{ s.t. } Y^\top Y = I,\ X^\top X = I. \tag{1}$$

The orthogonal relaxation is interesting because of its correspondence to hard (binary) clusterings: requiring that the columns of X and Y are orthogonal and nonnegative implies that every row has at most one nonzero entry.

Moreover, matrices having orthogonal columns may contain rows entirely filled with zeros, indicating that the corresponding observation or feature is not assigned to any cluster. However, orthogonal nonnegative matrix factorization is an NP-hard problem (Asteris et al. 2015), and in practice only soft-orthogonality of the matrices is achieved. On the one hand, the resulting fuzzy, soft-orthogonal indication of clusters is in principle suitable to model overlapping clusters as well. On the other hand, the fuzzy indication of clusters requires to make a-posteriori decisions to obtain definite cluster membership indications. A straightforward approach would be to assign observation j (or feature i) to the k clusters having the highest values in $Y_{j\cdot}$ (or $X_{i\cdot}$). Of course, imposing exclusivity would correspond to setting $k=1$ (Del Buono and Pio 2015; Yoo and Choi 2010), but determining the correct value of k for each observation or feature in an overlapping setting may be very problematic.

The optimization of (soft)-orthogonal factorizations is usually performed with multiplicative updates. These updates can be considered as gradient descent, where the step-size is chosen small enough such that the constraints are not violated. Unfortunately, the conservative choice of the step-size results in a slow convergence rate.

3.3 Methods based on penalization of nonbinary values

One of the very few attempts to solve binary constrained biclustering, without imposing the exclusivity, is the penalization approach proposed by Zhang et al. (2007, 2010, 2013). They propose a multiplicative update algorithm to minimize the approximation error together with a term that penalizes non-binary values. The employed penalization term is the Mexican hat function, shown in Fig. 2. Such a penalization scheme has also been applied for Boolean matrix factorization. This is a variant of binary biclustering, where clusters are explicitly allowed to overlap. Here, the matrix multiplication is computed in Boolean algebra, yielding $1 \oplus 1 = 1$. Note that, in binary matrix factorization, areas where two biclusters overlap are approximated by $1+1=2$, areas where three biclusters overlap are approximated by $1+1+1=3$, and so on. Hence, overlap of binary biclusters introduces an approximation error to the binary data matrix. In Boolean algebra, this is not the case and we always obtain a binary approximation by the Boolean product. In this context, Hess et al. (2017) proposed a proximal optimization scheme, employing the penalization function Λ defined in Eq. (3) and shown in Fig. 2. Adapting the function Λ instead of the Mexican Hat function has the advantage to lead to more matrix entries which are actually binary, instead of just being close to binary values. The optimization of a Boolean factorization in elementary algebra is approached by an optimization of the penalized objective and a subsequent thresholding to binary matrices. As we will detail in Sect. 4, Broccoli is built upon this approach, while exploiting the fact that the checkerboard biclustering objective does not require the approximation of a multiplication in another algebra.

3.4 Methods to explicitly model overlaps and outliers

In the literature, we can find a few methods which aim to specifically model outliers and overlaps among clusters. For example, Whang and Dhillon (2017) propose a semidefinite program that allows for a specified amount of overlap and a specified amount of outliers for row- and column-clusters. This approach introduces four parameters, which are not easy to estimate beforehand. Laclau and Brault (2019) propose a probabilistic binary biclustering model, which simultaneously facilitates feature selection. In this way, feature-(column-) clusters are not required to be exhaustive, and implicitly discard features that can be considered outliers. However, the same does not hold for the observations, and the identified clusters are still disjoint.

A specific context in which bicluster overlap has been explicitly modelled is that of bioinformatics. Among the approaches proposed in the literature, it is worth mentioning Floc (Yang et al. 2005), that addresses the issues originally present in the sequential approach proposed by Cheng and Church (2000). Floc is a probabilistic algorithm that can discover a set of possibly overlapping biclusters simultaneously, by iteratively and greedily determining the best actions (called *moves*) to perform for each row and column, as long as an improvement of the objective function is observed. Another relevant method is Fabia (Hochreiter et al. 2010), that is based on a multiplicative model that assumes that two feature vectors are similar if one is a multiple of the other, i.e., if the angle between them is zero or, as realization of random variables, their correlation coefficient is 1 or -1. This model was specifically adopted to capture possible linear dependencies between gene expressions and conditions, as well as heavy tailed distributions, which are typically observed in real-world transcriptomic data. However, although being generally able to discover overlapping biclusters, such approaches may appear to be inadequate if the underlying assumptions do not hold for the data at hand.

4 A novel stochastic optimization for non-exclusive biclusters

The objective of the biclustering task is nonconvex, as known for matrix factorizations. This entails that there are multiple local optima, which are typically not all suited to reflect a true, underlying clustering structure. Moreover, binary constraints on the matrices make this issue even more evident: every binary matrix induces a local optimum. Indeed, every binary matrix is the only feasible (and, therefore, the best) optimizer within its ϵ-ball neighborhood for small enough ϵ. Therefore, the optimization of the non-binary core matrix C, when fixing two arbitrary binary matrices, leads to a local optimum. It is noteworthy that a real-valued core matrix, as in the case of checkerboard biclustering, can lead to a significant decrease in the approximation error, even if the biclustering represented by the binary matrices is far away from the global optimum. This phenomenon makes it hard to distinguish between local optima and the global optimum by means of the objective function value (i.e., by observing the approximation error). In other words, having a *good* optimizer is not enough: We need a *very good* optimizer which simultaneously *(i)* handles the existence of many local optima that are almost indistinguishable from the global optimum by observing only the objective function, *(ii)* integrates binary constraints, and *(iii)* is robust to noise and can handle the presence of outliers.

4.1 Gradually increasing penalization of nonbinary values

We propose a biclustering optimization scheme based on the stochastic, proximal optimization framework SPRING (Driggs et al. 2020). Stochastic optimization computes the gradient descent updates in every iteration on a batch of the data. This makes stochastic optimization suitable for large scale data. Stochastic gradient descent is also known for its generalizing properties (Hardt et al. 2016; Hoffer et al. 2017).

We propose to optimize the following objective:

$$\min_{\substack{X,Y,C,\\ \boldsymbol{\lambda}_x,\boldsymbol{\lambda}_y}} \frac{1}{mn}\|D - YCX^\top\|^2 + \langle \boldsymbol{\lambda}_x, \Lambda(X) - \mathbf{1}\rangle + \langle \boldsymbol{\lambda}_y, \Lambda(Y) - \mathbf{1}\rangle + \phi_c(C)$$
$$\text{s.t. } \boldsymbol{\lambda}_x \in [0,\theta]^{n\times r}, \boldsymbol{\lambda}_y \in [0,\theta]^{m\times r}. \tag{2}$$

The parameter θ is here employed as a placeholder for the required regularization weights $\boldsymbol{\lambda}_x$ and $\boldsymbol{\lambda}_y$ such that the optimizing factor matrices Y and X of Eq. (2) are binary. Bounding the regularization weights above by the parameter θ ensures that the objective in Eq. (2) is well-defined. However, we do not need to determine the parameter θ in practice.

The regularization term $\langle \boldsymbol{\lambda}_x, \Lambda(X) - \mathbf{1}\rangle$ is composed of two parts: the nonbinary penalisation term $\langle \boldsymbol{\lambda}_x, \Lambda(X)\rangle$ and a term which pushes $\boldsymbol{\lambda}_x$ to be as large as possible $-\langle \boldsymbol{\lambda}_x, \mathbf{1}\rangle$. The matrix $\mathbf{1}$ is here a constant one matrix, whose dimensionality is inferred from the context. The parameter matrices $\boldsymbol{\lambda}_x$ and $\boldsymbol{\lambda}_y$ are the regularization weights of the non-binary penalization functions $\Lambda(X)$ and $\Lambda(Y)$. The matrix $\Lambda(X) = (\Lambda(X_{is}))$ is defined by the elementwise application of the function:

$$\Lambda(x) = \begin{cases} -|1-2x|+1 & x \in [0,1], \\ \infty & \text{otherwise.} \end{cases} \tag{3}$$

The function Λ (graphically depicted in Fig. 2) reaches its maximum value (1.0) at 0.5, its minimum value (0.0) at binary values, and returns infinity outside of the interval [0, 1]. It defines the feasible set of the matrices $X \in [0,1]^{n\times r}$ and $Y \in [0,1]^{m\times r}$ in the objective function [Eq. (2].
The Frobenius inner product

$$\langle \boldsymbol{\lambda}, \Lambda(X)\rangle = \sum_{i,s} \boldsymbol{\lambda}_{is}\Lambda(X_{is})$$

sums the elementwise penalization terms weighted by the parameters $\boldsymbol{\lambda}$.

The function Λ is non-smooth, but feasible for optimization by proximal gradient descent. Proximal optimization is a theoretically well-founded way to facilitate the optimization of non-smooth and possibly non-continuous terms of the objective. It is particularly used when the loss is a smooth function, and the constraints are possibly non-smooth but simple penalization terms. The proximal mapping of a function ϕ is a function which returns a matrix solving the following optimization problem:

$$\operatorname{prox}_{\phi}(X) \in \arg\min_{X^*} \left\{ \tfrac{1}{2} \|X - X^*\|^2 + \phi(X^*) \right\}. \tag{4}$$

Loosely speaking, the proximal mapping gives its argument a little push into a direction which minimizes ϕ. For a detailed discussion, see, e.g., the work by Parikh et al. (2014). This operator is employed in every iteration, which makes its optimization by numerical methods infeasible in practice. The trick is to use only simple functions ϕ for which the proximal mapping can be calculated analytically, in a closed form.

The proximal operator for Λ has been shown by Hess et al. (2017) to satisfy, for $x \in \mathbb{R}$

$$\operatorname{prox}_{\lambda\Lambda}(x) = \begin{cases} \max\{0, x - 2\lambda\} & x \le 0.5, \\ \min\{1, x + 2\lambda\} & x > 0.5. \end{cases} \tag{5}$$

The larger the regularization weight λ, the more the corresponding matrix value is pushed into the direction of binary values. After every gradient descent step of one of the cluster indicator matrices, the prox-operator is applied and pushes the matrix towards binary values. As a result, if we choose λ large enough, then we will get binary matrices after a couple of iterations. However, in this case, we also risk to converge to a local optimum close to the initialization. This would make our method even more sensitive to the initialization than it already is due to the nonconvexity of the objective. In turn, if we choose a too small value for λ, then the optimum of the penalized objective might not return binary matrices.

In order to circumvent these issues, we gradually increase the regularization weights throughout the optimization process. In addition, we employ individually determined regularization weights. To this end, we introduce the regularization weights as optimization parameters and subtract the sum of the regularization parameters $\langle \boldsymbol{\lambda}_x, \mathbf{1}\rangle + \langle \boldsymbol{\lambda}_y, \mathbf{1}\rangle$ from the objective function value. As a result, matrix entries which

Algorithm 1 The proposed general biclustering optimization scheme BROCCOLI, minimizing the objective in Eq (2).

```
function BROCCOLI($D, r, \gamma$)
    $(C, X, Y) \leftarrow$ INIT $(D, r)$
    $(\boldsymbol{\lambda}_X, \boldsymbol{\lambda}_Y) \leftarrow (0^{n \times r}, 0^{m \times r})$        ▷ Initialize the regularization weights of $\phi_X$ and $\phi_Y$
    $\mathcal{M}, \mathcal{N}, \mathcal{R} = \{1, \dots, m\}, \{1, \dots n\}, \{1, \dots, r\}$
    while not converged do
        Sample batches $\mathcal{I} \subseteq \mathcal{N}, \mathcal{J} \subseteq \mathcal{M}$
        $X \leftarrow$ UPDATE($X, \phi_x, \mathcal{M} \times \mathcal{I}$)        ▷ Note: $\phi_x(X) = \langle \boldsymbol{\lambda}_x, \Lambda(X) \rangle$
        $C \leftarrow$ UPDATE($C, \phi_c, \mathcal{M} \times \mathcal{I}$)
        $\boldsymbol{\lambda}_x \leftarrow \boldsymbol{\lambda}_x - \gamma(\Lambda(X) - \mathbf{1}) \restriction_{\mathcal{N} \times \mathcal{R}}$        ▷ Increase the penalization weights
        $Y \leftarrow$ UPDATE($Y, \phi_y, \mathcal{J} \times \mathcal{N}$)
        $C \leftarrow$ UPDATE($C, \phi_c, \mathcal{J} \times \mathcal{N}$)        ▷ Note: $\phi_y(Y) = \langle \boldsymbol{\lambda}_y, \Lambda(Y) \rangle$
        $\boldsymbol{\lambda}_y \leftarrow \boldsymbol{\lambda}_y - \gamma(\Lambda(Y) - \mathbf{1}) \restriction_{\mathcal{M} \times \mathcal{R}}$        ▷ Increase the penalization weights
    end while
    return $(C, X, Y)$
end function
function UPDATE($A, \phi, \mathcal{B}$)
    $\nabla_A \leftarrow \nabla_A \text{MSE}(C, X, Y) \restriction_{\mathcal{B}}$        ▷ Batch gradient
    $\alpha \leftarrow 1/4L_{\nabla_A}$
    return $\text{prox}_{\alpha\phi}(A - \alpha\nabla_A)$        ▷ Proximal gradient step
end function
```

naturally fall close to a binary value receive a stronger push towards binary values than those entries which are *undecided* (i.e., close to 0.5).

4.2 The method BROCCOLI

We formally describe our method BROCCOLI in Algorithm 1. The method returns a non-exclusive biclustering model, which is defined by the specification of constraints on the core matrix by ϕ_c. In this paper, we focus on the identification of nonnegative checkerboard clusterings. Specifically, we set

$$\phi_c(x) = \begin{cases} \infty, & \text{if } x < 0 \text{ or } x > max_c \\ 0, & \text{otherwise.} \end{cases}$$

The proximal operator $\text{prox}_{\alpha\phi_c}(C)$ projects the elements of the matrix C onto the interval $C_{s,t} \in [0, max_c]$, with the effect of bounding the maximum value of C. This prevents an imbalance of the matrices where X or Y have very low values, which are compensated by large values in C.

The input of our method BROCCOLI is the data matrix D, the rank of the factorization r and the step-size γ of the gradient descent steps, updating the regularisation parameters $\boldsymbol{\lambda}_x$ and $\boldsymbol{\lambda}_y$.

In every iteration, a batch of columns of the data matrix is sampled and the matrices C and X, as well as the regularization weights, are updated. Likewise, a batch of rows of the data matrix is sampled subject to which the matrices C and Y are updated. Every update computes the proximal mapping of the gradient step, performed on the respective batch. The step-size is chosen by means of the Lipschitz constant of the gradient (cf. Appendix A). Under these circumstances, SPRING guarantees convergence to a local minimum in expectation (Driggs et al. 2020).

4.2.1 Initialization

As generally known from matrix factorization, the initialization may influence the results. There are multiple issues which can occur during the optimization of matrix factorizations with binary constraints, and which can be alleviated with a good initialization method. One of these issues is an imbalance in the scale of the factor matrices (Zhang et al. 2007). For nonnegative matrix factorization this is not a big issue. However, it is a problem when binary constraints are imposed. Suppose that the ground truth biclustering is given by the matrices Y^*, X^* and C^*. Let $\mathbf{x}, \mathbf{y} \in \mathbb{R}^r$ be positive vectors. Then, for $Y = Y^* \operatorname{diag}(\mathbf{y})^{-1}$, $C = \operatorname{diag}(\mathbf{y})C^* \operatorname{diag}(\mathbf{x})$ and $X = X^* \operatorname{diag}(\mathbf{x})^{-1}$, we have:

$$Y^*C^*X^{*\top} = Y^* \operatorname{diag}(\mathbf{y})^{-1} \operatorname{diag}(\mathbf{y})C^* \operatorname{diag}(\mathbf{x}) \operatorname{diag}(\mathbf{x})^{-1} X^{*\top} = YCX^\top. \tag{6}$$

As a result, the matrix product $YCX^\top$ is indistinguishable from the one of the ground truth, although the factor matrices differ from the ground truth factors. During optimization, when the binary constraints are relaxed, we might obtain differently scaled matrices as iterates. If the vector $\mathbf{x}$ in Eq. (6) contains large numbers, then the matrix X has columns which are close to zero. In upcoming iterations, the proximal operator pushes these small values even closer to zero, which has to be balanced by the scaling of C. This scaling coping mechanism works until the values in X reach zero. If that is the case, then we observe a sudden increase of the approximation error.

The scaling issue can be alleviated by a suitable initialization. We consider two possible schemes (see Algorithm 2): a baseline approach consisting of a uniformly random initialization of the (relaxed) cluster indicator matrices (henceforth this approach will be denoted as InitRND), and a more sophisticated initialization based on a shortly optimized nonnegative matrix factorization (henceforth this approach will be denoted as InitNMF). The latter approach is the one that we propose in this paper. In both cases, the inputs are the data matrix D and the rank r. The method InitNMF has one additional parameter p, specifying the percentage of fuzzy cluster assignments which are set to 1.0 after the initialization. In Broccoli we adopt $p = 80$, which according to a set of independent preliminary experiments, appears to be appropriate and reasonable.

The proposed InitNMF employs the uniformly random initialization InitRND for its own initialization. The optimization consists of 100 proximal stochastic gradient descent updates for nonnegative matrix factorization. The nonnegative constraints are incorporated by the regularizing function ϕ_+, which returns infinity at negative matrix entries, zero otherwise. The proximal operator of this function is a projection of the argument matrix onto nonnegative values (Bolte et al. 2014).

Given the resulting nonnegative factors (X_+, Y_+), they are *converted* into a suitable tri-factorization (C, X, Y). Ideally, the initialization yields factor matrices X and Y which already reflect the optimal distribution of zeros and ones per cluster. Since the optimal distribution is not known in advance, we employ a heuristic strategy.

Algorithm 2 Proposed NMF and RND initialization schemes for BROCCOLI.

```
1: function InitNMF($D, r; p = 80$)
2:     $(C, X, Y) \leftarrow$ InitRND $(D, r)$
3:     $\mathcal{M}, \mathcal{N} = \{1, \dots, m\}, \{1, \dots n\}$
4:     for $t \in \{1, \dots 100\}$ do
5:         Sample batches $\mathcal{I} \subseteq \mathcal{N}, \mathcal{J} \subseteq \mathcal{M}$
6:         $X \leftarrow$ Update($X, \phi_+, \mathcal{M} \times \mathcal{I}$)          ▷ Note: $\phi_+(x) = \infty$ if $x < 0$
7:         $Y \leftarrow$ Update($Y, \phi_+, \mathcal{J} \times \mathcal{N}$)
8:     end for
9:     $\mathbf{x} \leftarrow (P_{p\%}(X_{\cdot 1}), \dots, P_{p\%}(X_{\cdot r}))$
10:    $\mathbf{y} \leftarrow (P_{p\%}(Y_{\cdot 1}), \dots, P_{p\%}(Y_{\cdot r}))$
11:    $(\mathbf{x}_s, \mathbf{y}_s) \leftarrow \left(\mathbf{x}_s + 0.1\frac{\mathbf{x}_s}{\sqrt{\mathbf{x}_s \mathbf{y}_s}},\ \mathbf{y}_s + 0.1\frac{\mathbf{y}_s}{\sqrt{\mathbf{x}_s \mathbf{y}_s}}\right)$ for all $1 \leq s \leq r$
12:    $C \leftarrow [\mathrm{diag}(\mathbf{x})\,\mathrm{diag}(\mathbf{y})]_{\leq max_c}$
13:    $(Y, X) \leftarrow \left(\left[Y\,\mathrm{diag}(\mathbf{y})^{-1}\right]_{\leq 1}, \left[X\,\mathrm{diag}(\mathbf{x})^{-1}\right]_{\leq 1}\right)$
14:    return $(C, X, Y)$
15: end function
16: function InitRND($D, r$)
17:    $C \rightarrow I$          ▷ $C$ is equal to the $r \times r$ identity matrix
18:    $X \rightarrow$ SampleUniform($m, r, [0, 1]$)
19:    $Y \rightarrow$ SampleUniform($m, r, [0, 1]$)
20:    return $(C, X, Y)$
21: end function
```

In particular, we determine scaling vectors $\mathbf{x}$ and $\mathbf{y}$, whose product will reflect the diagonal entries of C (cf. Steps 12 and 13). In the first step, we set the scaling vectors $\mathbf{x}$ and $\mathbf{y}$ to the pth percentile of the nonnegative indicators of each cluster. In this way, we ensure that the cluster indicator matrices are not too sparse, having at least $100 - p\%$ of all entries equal to one. At the same time, we need to ensure that the core matrix C is not too sparse. If one row or column of the core matrix is equal to zero, then the corresponding row- or column-cluster is not used. Therefore, we add 0.01 to the diagonal of the core matrix. This is achieved by adding 0.1 to the scaling vectors, where we additionally apply a weighting scheme (cf. Step 11) that provides a higher weight to denser factor matrices than to sparse indicator matrices. BROCCOLI's tri-factorization is finally initialized by the scaled nonnegative matrix factorization.

4.2.2 Time complexity analysis

The complexity of BROCCOLI's optimization scheme corresponds to the complexity of an update step times the number of required iterations. In the following, we derive the complexity of each update step as $\mathcal{O}(mnr)$, resulting in an overall time complexity of $\mathcal{O}(Tmnr)$, where T is the number of iterations.

For each update step, we compute the gradient, the Lipschitz constant and the proximal operator. The complexity for computing the gradient and the Lipschitz constant can be derived by the complexity of matrix multiplications, that is $\mathcal{O}(mnr)$ for the multiplication of an $n \times r$ matrix with an $r \times m$ matrix. Accordingly, the gradients are computed in $\mathcal{O}(mnr)$, while the Lipschitz constant is computed in $\mathcal{O}(r^2)$, if the required matrix products are reused from the computation of the gradient (cf. Appendix A). Finally, all employed proximal operators require constant time to update one matrix entry [cf. Eq. (5)]. Therefore, the proximal mappings of a factor matrix are computed in $\mathcal{O}(\max\{nr, mr, r^2\})$. Since we can assume that $r \leq \min\{n, m\}$, the dominating factor is the computation of the gradient, resulting in an overall complexity for each update step equal to $\mathcal{O}(mnr)$.

As regards the initialization strategy, it is noteworthy that it does not affect the time complexity, since the more expensive initialization scheme proposed in this paper (InitNMF) takes $\mathcal{O}(mnr)$, required by the gradient update steps. In this case, the number of iterations is constant ($T = 100$) and does not asymptotically affect the time complexity.

5 Experiments

We compare the proposed method Broccoli against six competitors: two methods based on a nonnegative relaxation (henceforth denoted by N (Long et al. 2005) and NN (Del Buono and Pio 2015)), two methods based on an orthogonal relaxation (henceforth denoted by O (Yoo and Choi 2010) and OO (Del Buono and Pio 2015)) and the biclustering methods Fabia (Hochreiter et al. 2010) and Floc (Yang et al. 2005). Since N, NN, O and OO return fuzzy membership values for each observation, we binarize the result for comparison purposes. For each sample (observation or feature) we set the top-k fuzzy cluster indicator values to one, where k is the number of ground truth clusters the sample belongs to. Note that in this way we provide our competitors with additional background knowledge, which is not available in real-world scenarios. The goal is to estimate how good the clustering, derived from a relaxed result, could potentially be, if supported by additional knowledge provided (e.g., by domain experts).

Broccoli[1] is implemented in PyTorch, exploits the inbuilt implementation of SGD and batch sampling, and relies on the heavily parallelized execution of matrix multiplications by Graphics Processing Units (GPUs). The batch sampling is performed epoch-wise. In every epoch, data is partitioned into 10 sets, which are returned as batches in subsequent iterations. Hence, the size of the batches are approximately equal to $0.1n$ and $0.1m$, respectively. As default values for the step-size of the regularization weight updates, we set $\gamma = 10^{-9}$, that is suitable for most datasets. In practice, the step-size should be as low as the run-time allows. In order to speed up convergence without noticeably harming the quality of the results, we double the step-size γ every 2000 epochs. We set the parameter max_c of the function ϕ_c equal to the maximum value in D.

The rank is set equal to the rank of the ground truth. In some experiments we denote the ranks r_y and r_x, which are the number of row-clusters and column-clusters, respectively, of the obtained result. Note that the ranks of the returned tri-factorization may be smaller than the specified rank, due to constant zero columns in X or Y, or constant zero columns or rows in C.

[1] The system is publicly available at: https://github.com/Sibylse/Broccoli.

We evaluate BROCCOLI using both the initialization schemes described in Sect. 4.2.1. Specifically, we denote the variant based on INITNMF as BROCCOLI NMF, and the variant based on INITRND as BROCCOLI RND.

5.1 Evaluation measures

We quantify how well a computed cluster indicator matrix Y matches the ground truth Y^* by an adaptation of the averaged F_1-measure, known from multi-class classification tasks. We compute a one-to-one matching τ between computed and ground truth clustering and compute the average F_1-measure of the matched clusters. Formally, the F_1-measure of two binary vectors y and y^* is computed as the harmonic mean of precision and recall:

$$\mathrm{pre}(y, y^*) = \frac{y^\top y^*}{\|y\|^2}, \quad \mathrm{rec}(y, y^*) = \frac{y^\top y^*}{\|y^*\|^2}.$$
$$F_1(y, y^*) = 2\frac{\mathrm{pre}(y, y^*) \cdot \mathrm{rec}(y, y^*)}{\mathrm{pre}(y, y^*) + \mathrm{rec}(y, y^*)} = \frac{2y^\top y^*}{\|y\|^2 + \|y^*\|^2}.$$

The average F_1-measures for column- and row-clusters are then computed by

$$F_1(Y, Y^*) = \frac{1}{r_y} \sum_{s=1}^{r_y} F_1(Y_{\cdot s}, Y^*_{\cdot \tau_y(s)}), \qquad F_1(X, X^*) = \frac{1}{r_x} \sum_{t=1}^{r_x} F_1(X_{\cdot t}, X^*_{\cdot \tau_x(t)}).$$

In addition to this matching-based measure, we also compute two agreement measures for overlapping clusterings proposed by Rabbany and Zaïane (2015). Given a cluster indicator matrix Y, these measures are defined as

$$I_{\cos}(Y, Y^*) = \frac{\|Y^\top Y^*\|^2}{\|Y^\top Y\| \|Y^{*\top} Y^*\|} \qquad I_{\mathrm{sub}}(Y, Y^*) = \frac{\|Y^\top Y^*\|}{\|Y\| \|Y^*\|}.$$

The term

$$\|Y^\top Y^*\|^2 = \sum_{1 \le s,t \le r} (Y_{\cdot s}^\top Y_{\cdot s}^*)^2 = |YY^\top \circ Y^* Y^{*\top}|$$

represents the agreement according to the number of elements that the two clusters $Y_{\cdot s}$ and $Y_{\cdot t}$ have in common (see Appendix B for a more detailed discussion of the clustering agreement indexes).

If we are given the ground truth for the row- and column-clusters, then we return the average of the measures:

$$F_1 = \tfrac{1}{2}(F_1(Y, Y^*) + F_1(X, X^*))$$
$$I_{\cos} = \tfrac{1}{2}(I_{\cos}(Y, Y^*) + I_{\cos}(X, X^*))$$
$$I_{\mathrm{sub}} = \tfrac{1}{2}(I_{\mathrm{sub}}(Y, Y^*) + I_{\mathrm{sub}}(X, X^*))$$

Table 2 Statistics about the synthetic datasets

m	n	Clusters	Overlap X	Overlap Y	Outliers X	Outliers Y
300	200	3	1.17 ± 0.03	1.17 ± 0.03	$44 \pm 3\%$	$43 \pm 3\%$
1000	800	5	1.33 ± 0.02	1.33 ± 0.02	$25 \pm 1\%$	$24 \pm 1\%$

All measures range between 0.0 and 1.0. The closer they approach 1.0, the more the computed clustering matches the ground truth. Note that I_{sub} does generally attain a value smaller than 1.0 also if the cluster indicator matrix perfectly corresponds to the ground truth.

Finally, we report the value of the approximation error of the factorization with respect to the original data matrix, measured through the Mean Squared Error in percentage (MSE%). Formally:

$$\text{MSE\%} = \frac{\|D - YCX^\top\|^2}{\|D\|^2} \cdot 100.$$

Note that, contrary to the performance measures, the lower the MSE% the better the approximation. After all, the MSE% represents the average error made by the factorization of the input data matrix. However, we also caution that a low approximation error does not necessarily indicate a correctly identified clustering structure (cf. Sect. 4).

5.2 Synthetic datasets

We create a set of synthetic datasets with overlap and outliers by sampling every cluster indicator matrix by a Bernoulli distribution. Each entry X^*_{it} and Y^*_{js} is equal to 1 with probability $q = 0.2$. Thereby, we ensure that each cluster contains at least 1% of the observations/features, which are exclusively assigned to that specific cluster. The core matrix is sampled as a sparse matrix containing uniformly distributed values $C_{st} \in [0, 5]$. The probability that a non-diagonal element is not zero is equal to $1/r$. The data matrix is then generated by adding random Gaussian noise to the ground truth factorization:

$$D_{ji} = [Y^*_{j\cdot} C X^{*\top}_{i\cdot} + \epsilon_{ji}]_{\geq 0},$$

where $\epsilon_{ji} \sim \mathcal{N}(0, \sigma)$ and the operator $[\cdot]_{\geq 0}$ projects negative values to zero. We generate five datasets for every noise variance $\sigma \in \{0, 0.2, 0.4, \ldots, 2\}$ and dimensionality $(m, n) \in \{(300, 200), (1000, 800)\}$ (see Table 2 for a summary of the statistics of the generated datasets).

For the smaller 300×200 dataset we choose a rank of $r = 3$, while for the larger 1000×800 dataset we choose a rank of $r = 5$. The characteristics *overlap X* and *overlap Y* denote the average number of clusters a feature or observation is assigned to, when it is not an outlier. Outliers are features or observations which are not assigned to any cluster.

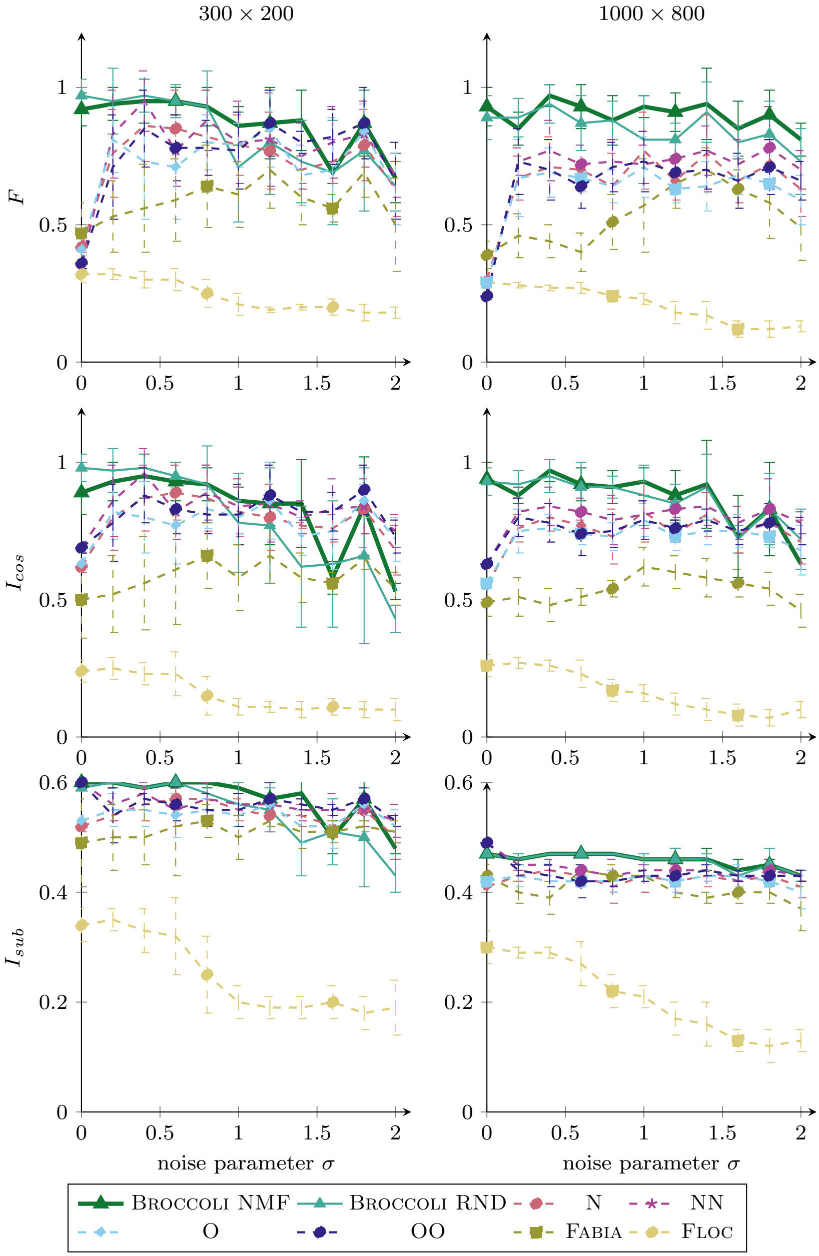

Fig. 3 Average F_1-, I_{cos}- and I_{sub}-measure (the higher the better), plotted against the Gaussian noise parameter σ. Error bars have a length of twice the standard deviation

Table 3 Characteristics of selected datasets for multi-label classification

D	m	n	Classes	Overlap Y	Outliers Y	$D_{ji} \in$	$P_{99.9\%}$
Birds	645	260	19	1.86 ± 0.97	45%	[0, 103512]	3806.7
Emotions	593	72	6	1.86 ± 0.67	0%	[0, 336]	299.1
Genbase	662	1185	19	1.23 ± 0.67	0%	{0, 1}	1.0
Scene	2407	294	6	1.07 ± 0.26	0%	[0, 1]	1.0
Yeast	2417	103	14	4.24 ± 1.57	0%	[0, 1.52]	1.21
20News	11314	6643	20	0.00±0.00	0%	[0, 800]	3.0

We denote the dimensionalities m, n, the number of classes, the average number of classes an observation is assigned to (overlap Y), the percentage of outliers, the range of values in the data matrix and the 99.9th percentile of values in the data matrix

In Fig. 3, we plot the F_1-, I_{cos}- and I_{sub}-measures, against the Gaussian noise parameter σ. For $\sigma = 2$, roughly $1/3$ of the noise samples are larger than or equal to 1.0, and about $2/3$ of all noise samples have an absolute value larger than or equal to 1.0 in expectation. We can observe that the measures overall indicate a similar ranking of the performance of the algorithms. Throughout the increase of the noise, Broccoli NMF attains very high scores, while Broccoli RND exhibits slightly lower performances. We also observe an increased variance of Broccoli RND's results (shown by the error bars), indicating a sensitivity to the random initialization. Broccoli's performance drops especially in the I_{cos}-measure when the noise exceeds 1.5. Since this drop is more pronounced in the small dataset (300×200), it may be due to the combination of a high amount of noise and of a high number of outliers.

The methods N, NN, O and OO, which are based on orthogonal and nonnegative relaxations, seem largely unaffected by the noise parameter. We remind that we provided these methods with the advantage of knowing the true number of clusters for each observation or feature, during the binarization. On the contrary, this advantage was not provided to Broccoli. Despite such an advantage, they never achieve a score larger than 0.8. Fabia and Floc attain the lowest scores among all the considered competitors, where the scores of Fabia have a tendency to increase with the noise parameter. This behavior is possibly due to the fact that Fabia (together with Floc) does not explicitly handle the possible presence of noise in the data, and is therefore very sensitive to it. The difficulty faced in modeling the true clustering structure with no noise (F_1- and I_{cos}-measure close to 0.5 for $\sigma = 0$) may ideally be alleviated in presence of a huge amount of noise, which strongly pushes it away from the (wrong) local minimum it was possibly stuck on.

5.3 Real-world datasets

We also evaluate our method on a series of real-world datasets, originally designed for multi-label classification tasks. These datasets have naturally multiple classes per observation, which we employ as the ground truth for evaluation purposes. The statistics of these datasets are depicted in Table 3. The *birds* dataset is derived from audio files (Briggs et al. 2013), *emotions* addresses the sentiment of music (Trohidis et al.

2008), *scene* is an image dataset (Boutell et al. 2004) and *genebase* and *yeast* are derived from the biological domain (Diplaris et al. 2005; Elisseeff and Weston 2002). We also adopt the *20 Newsgroups (20News)* dataset[2], that in principle is not multi-label. However, since in *20News* labels are hierarchically organized, it allows us to emphasize the capabilities of the considered methods in catching overlaps in terms of hierarchically organized clusters (i.e., a more general category is overlapping with—actually includes—more specific categories).

We evaluate how well the computed row-clusters match with the given labels. However, we should keep in mind that the fundamental task of clustering is to find the *prevalent* structure in the dataset, which does not necessarily correspond to *the specific* structure encoded by the class labels.

Table 4 summarizes the results on the selected multilabel datasets. As a benchmark, we train the factor matrix X of a nonnegative matrix factorization when fixing Y to the class-assignment matrix:

$$\min_X \frac{1}{nm}\|D - YX^\top\|^2, \quad \text{s.t. } X \in \mathbb{R}_+^{n\times r}, \tag{7}$$

The optimization problem in Eq. (7) is convex. Hence, we can compute the global minimum of this objective, which provides a lower bound of the MSE which could possibly be attained by a biclustering where the row-clusters actually reflect the classes. We refer to this method as *NMF given Y*.

Table 4 does not display MSE% values for the algorithms FABIA and FLOC, since these algorithms only return the cluster indicator matrices and not the core matrix. The performance measures of N, NN, O and OO are again derived from a binarization of the fuzzy row-clusters by means of the class-indicator matrix. The MSE of these methods is denoted for the relaxed (fuzzy) factor matrices. For all methods we set the number of clusters r equal to the number of classes.

Table 4 displays how well the competing algorithms match the given class labels. For five out of the six datasets, the row-clustering of BROCCOLI NMF attains the best score in at least one of the considered measures.

We observe that BROCCOLI (NMF and RND) returns fewer clusters with respect to the specified rank, for *birds* and *genbase*. In this case, we also report in the parentheses the F_1-score averaged only over the best-matched clusters identified by BROCCOLI. In other words, the F_1-scores in parentheses denote how well BROCCOLI's clusters match a selection of true classes.

The *birds* dataset poses an exception for BROCCOLI NMF. First, BROCCOLI NMF is not able to factorize this dataset with a low MSE%. This is partly due to the fact that almost half of the observations do not belong to any class and are thus marked as outliers. Additionally, the gap between the 99.9th percentile of data values and the maximum data value is very big (cf. Table 3). Therefore, there are very few values which dominate the mean squared error if these values are not approximated. In this case, the relaxed methods (O,OO, N and NN) are in advantage, because their relaxed approximation scheme allows for the adaptation to single, exceptionally high values

[2] http://qwone.com/~jason/20Newsgroups/.

Table 4 Evaluation of the clustering methods on real-world datasets

Data	Method	MSE %	$F(Y, Y^*)$	$I_{\cos}(Y, Y^*)$	$I_{\mathrm{sub}}(Y, Y^*)$	r_X	r_Y
Birds	NMF given Y	62.57	1.00	1.00	0.32	–	19
	Broccoli NMF	72.18	0.11 (0.12)	0.28	**0.29**	11	17
	Broccoli RND	85.23	0.09 (0.16)	0.27	**0.29**	7	11
	N	0.05	0.20	0.42	0.26	19	19
	NN	0.04	0.20	0.38	0.23	19	19
	O	0.14	0.21	**0.43**	0.24	19	19
	OO	0.08	**0.23**	0.37	0.20	19	19
	Fabia	–	0.20	0.32	0.20	9	19
	Floc	–	0.10	0.25	0.28	19	19
Emotions	NMF given Y	9.30	1.00	1.00	0.51	–	6
	Broccoli NMF	0.05	**0.52**	0.64	**0.50**	6	6
	Broccoli RND	0.05	0.51	**0.65**	0.48	6	6
	N	0.13	0.40	0.56	0.36	6	6
	NN	0.14	0.38	0.56	0.36	6	6
	O	0.16	0.37	0.59	0.39	6	6
	OO	0.11	0.37	0.59	0.43	6	6
	Fabia	–	0.47	0.46	0.30	3	6
	Floc	–	0.42	0.56	0.44	6	6
Genbase	NMF given Y	0.07	1.00	1.00	0.33	–	19
	Broccoli NMF	0.27	0.09 (0.57)	**0.37**	**0.33**	19	3
	Broccoli RND	97.56	0.01 (0.19)	0.03	0.10	19	1
	N	0.20	0.16	0.14	0.11	19	19
	NN	0.21	0.16	0.15	0.11	19	19
	O	0.20	0.18	0.17	0.13	19	19
	OO	0.20	0.20	0.16	0.13	19	19
	Fabia	–	**0.29**	0.29	0.23	17	19
	Floc	–	0.11	0.28	0.30	19	19
Scene	NMF given Y	19.56	1.00	1.00	0.41	–	6
	Broccoli NMF	16.20	**0.45**	**0.48**	**0.32**	6	6
	Broccoli RND	20.09	0.27	0.35	0.26	6	6
	N	15.08	0.33	0.35	0.26	6	6
	NN	12.75	0.43	0.40	0.27	6	6
	O	15.04	0.37	0.33	0.25	6	6
	OO	15.67	0.37	0.30	0.23	6	6
	Fabia	–	0.43	0.39	0.30	6	6
	Floc	–	0.42	0.22	0.21	6	6

Table 4 continued

Data	Method	MSE %	$F(Y, Y^*)$	$I_{\cos}(Y, Y^*)$	$I_{\text{sub}}(Y, Y^*)$	r_X	r_Y
Yeast	NMF given Y	13.24	1.00	1.00	0.56	–	14
	Broccoli NMF	1.38	0.29	0.57	0.32	14	14
	Broccoli RND	20.09	0.27	0.35	0.26	14	14
	N	1.49	0.32	0.80	0.42	14	14
	NN	1.39	0.34	0.80	0.42	14	14
	O	1.49	0.36	**0.82**	**0.45**	14	14
	OO	1.40	**0.35**	0.81	0.44	14	14
	Fabia	–	0.34	0.63	0.37	14	14
	Floc	–	0.05	0.06	0.10	14	14
20News	NMF given Y	95.36	1.00	1.00	0.22	–	20
	Broccoli NMF	93.18	**0.14**	0.07	0.08	20	20
	Broccoli RND	98.77	0.01	0.00	0.02	20	20
	N	55.70	0.04	0.20	0.18	20	20
	NN	56.10	0.04	0.21	0.21	20	20
	O	58.30	0.02	**0.22**	**0.22**	20	20
	OO	57.06	0.05	**0.22**	**0.22**	20	20
	Fabia	–	-	–	–	–	–
	Floc	–	–	–	–	–	–

We compare the novel method Broccoli with the selected competitors and a baseline NMF, where we fix the cluster assignment matrix Y to the class-assignment matrix and optimize only for one other factor matrix (NMF given Y). Best results are emphasized in bold, except for the (lowest) MSE%, since the values are not directly comparable (all MSE% values, except for those measured for Broccoli, are computed based on a relaxed solution). The F_1-scores reported in the parentheses correspond to the average F_1-scores considering only the best-matched clusters identified by Broccoli

within one bicluster (a similar effect is observable on *20News*). In addition, the binarization ensures that all the outliers are also reflected as such in the binary factorization, which positively influences the performance measures.

In contrast, the *genbase* dataset is a binary dataset which is similarly well-factorized by Broccoli NMF as by the relaxed methods. In particular, Broccoli NMF attains a low MSE% with only three of the possible 19 row-clusters. With these few clusters, Broccoli NMF achieves the highest $I_{\cos}$ and I_{sub} agreement measures, which do not explicitly depend on the number of modeled clusters due to their scaling invariance (cf. Appendix B). Yet, Broccoli NMF and Broccoli RND produce the lowest F_1-score, which is largely due to the fact that the average F_1-score returns a score of zero for every non-modeled cluster. On the other hand, the high F_1-score (shown in the parentheses), averaged over only the modeled three clusters, indicates a proper match of the clusters modeled by Broccoli NMF with true labels.

The *genbase* dataset is the only one for which Fabia attains a high score in at least one of the measures. On the other datasets, Fabia is often reaching good but not outstanding scores, which are not too far from the best ones. Floc usually obtains lower F_1-scores than Fabia, but the results in terms of agreement measures are discordant.

Table 5 Summary of the *20 Newsgroup* categories

Belief	Comp	Misc	Rec	Sci	Politics
alt.atheism	graphics	forsale	autos	crypt	guns
religion.christian	os.ms-windows.misc		motorcycles	electronics	mideast
talk.religion	sys.ibm.pc.hardware		sport.baseball	med	misc
	sys.mac.hardware		sport.hockey	space	
	windows.x				

The topics are divided into six categories which have a set of sub-categories

We observe that for all the other datasets, Broccoli NMF attains a lower MSE% than the benchmark NMF for given Y. This shows that the true class labels generally do not yield a suitable clustering, which approximates the data well. No clustering which actually aligns with the class labels can obtain a lower MSE% than the one denoted for NMF given Y. After all, the MSE% measures how well the identified clustering structure matches the data. Hence, any clustering which obtains a MSE% which is substantially lower than the baseline of NMF given Y cannot have a performance measure close to 1.0. We further note that Broccoli NMF usually attains a MSE% which is close to the one of relaxed factorizations N, NN, O and OO. For the *emotions* and the *yeast* dataset, Broccoli achieves even a lower MSE% than the relaxed factorizations. This demonstrates the strengths of the employed optimization.

Finally, we observe that Broccoli RND, based on a random initialization, generally performs worse than Broccoli NMF. In particular, we can observe significantly higher values of the MSE%, indicating that Broccoli RND often converges to noticeable worse local optima than Broccoli NMF. This result confirms the positive contribution of the proposed initialization strategy that, on the other hand, requires a negligible amount of additional running time.

5.4 Qualitative evaluation of the 20 newsgroups dataset

We perform a qualitative evaluation of results on the *20News* dataset. This dataset is a collection of posts assigned to 20 topics which are hierarchically organized (cf. Table 5). We process the textual data as a data matrix, corresponding to the term frequency of $n = 6643$ lemmatized words in $m = 11314$ posts (training data only, as provided by scikit-learn[3]). We apply NN, OO, Broccoli RND and Broccoli NMF to derive $r = 20$ row- and column-clusters. Fabia and Floc were not able to process such a large dataset.

5.4.1 Inspection of feature clusters

The obtained column-clusters (the feature clusters which, in this case, are clusters of words) are shown in Figs. 4, 5, 6 and 7. For the fuzzy cluster indicators of NN and OO, the size of the word i in the wordcloud of cluster s corresponds to the assigned weight

[3] https://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_20newsgroups.

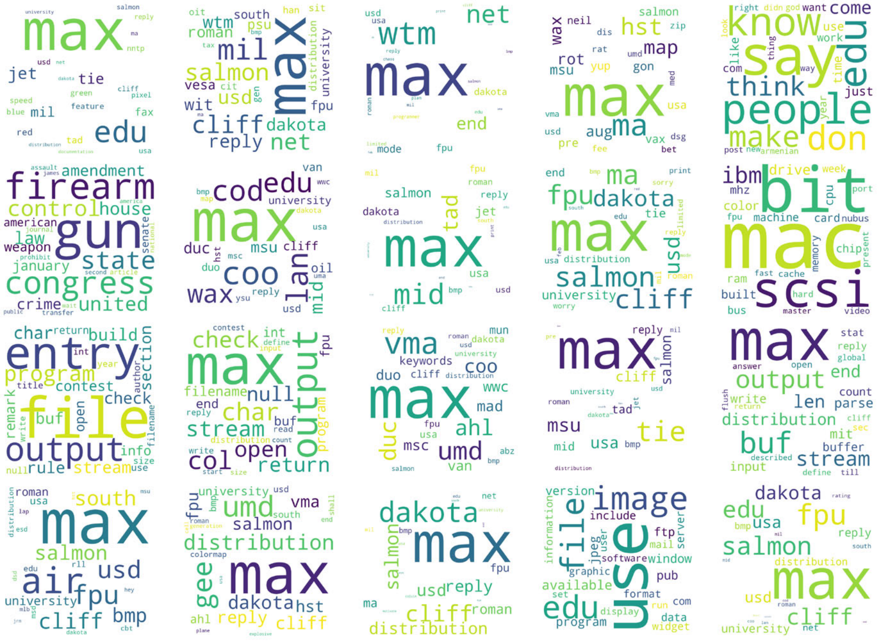
max jet tie edu mil usd salmon cliff fax
max mil salmon cliff reply net dakota usd wtm roman south
max wtm net end dakota mode fpu
max hst map ma rot msu neil salmon end usa
know say people think make don edu come just
firearm gun control house congress state united crime law weapon january amendment american
max cod edu coo wax lan mid duc oil msu cliff
max mid salmon tad dakota jet reply usa
max salmon cliff fpu dakota ma usd university distribution
mac bit scsi ibm drive chip ram built bus
file entry output program check buf rule stream char build contest
max output check null char stream col open return filename
max vma duc umd ahl coo wwc mad keywords
max tie msu usa mid tad jet reply salmon
max output buf stream distribution write len end count buffer
max air south salmon usd fpu cliff bmp university
max umd distribution gee dakota hst reply cliff salmon
max dakota salmon usd reply cliff distribution roman
use image file edu ftp include available format
max salmon fpu edu cliff dakota usa university

Fig. 4 Illustration of derived word-clusters by the method NN on the 20 Newsgroups dataset. The size of a word reflects its weight in the corresponding cluster ($X_{\cdot s}$)

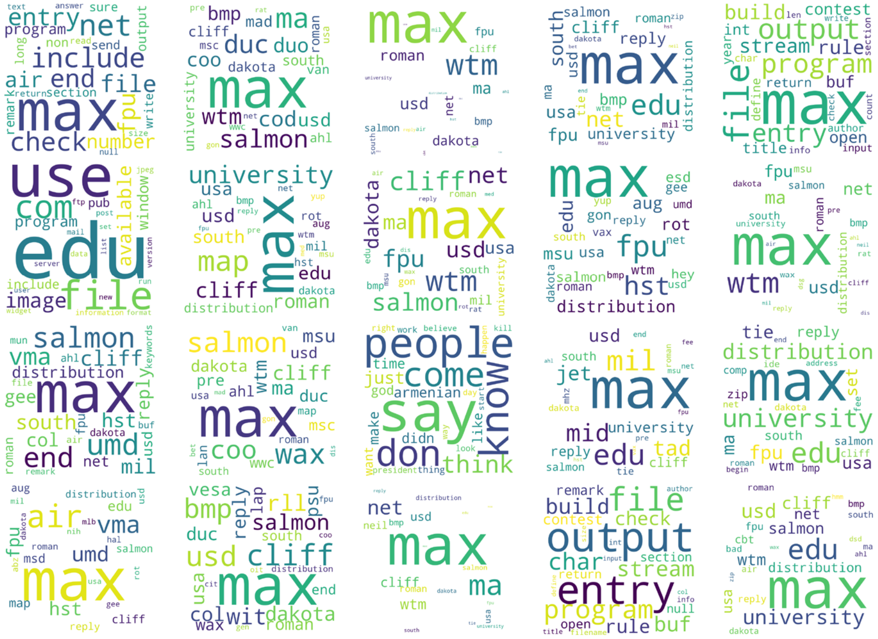
max entry net include file check number air end fpu
max duc salmon wtm cod usd bmp cliff ma coo
max wtm roman usd fpu net dakota salmon
max edu net usa fpu university cliff salmon reply
max output file program stream entry rule build contest
use edu com file image available program
max university usd map cliff south roman edu
max cliff dakota net ma usd fpu wtm salmon
max edu fpu hst gon aug rot reply distribution
max fpu wtm net ma usd
max salmon cliff vma distribution gee end south umd mil
max salmon coo ahl wax usd cliff duc msu
people come say don know think just time
max mil jet mid usd edu tad reply
max tie distribution university edu fpu wtm usa
max air vma umd fpu hst edu
max cliff usd bmp reply salmon col wit dakota
max net usd cliff ma wtm
file output entry program char build check stream buf
max usd cliff net salmon edu university distribution

Fig. 5 Illustration of derived word-clusters by the method OO on the 20 Newsgroups dataset. The size of a word reflects its weight in the corresponding cluster ($X_{\cdot s}$)

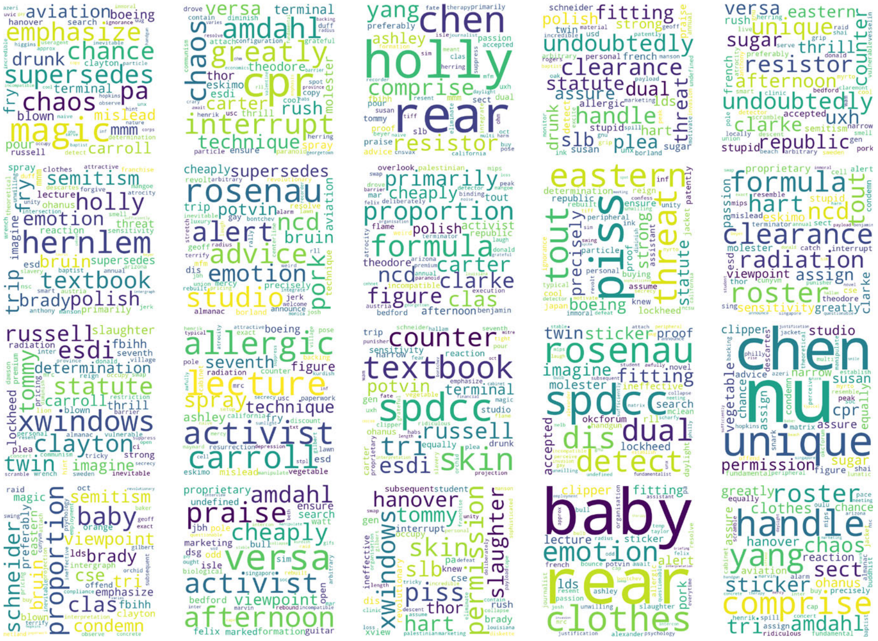

Fig. 6 Illustration of derived word-clusters by BROCCOLI RND on the 20 Newsgroups dataset. The size of a word reflects its weight in the corresponding cluster $(X_{\cdot s})$

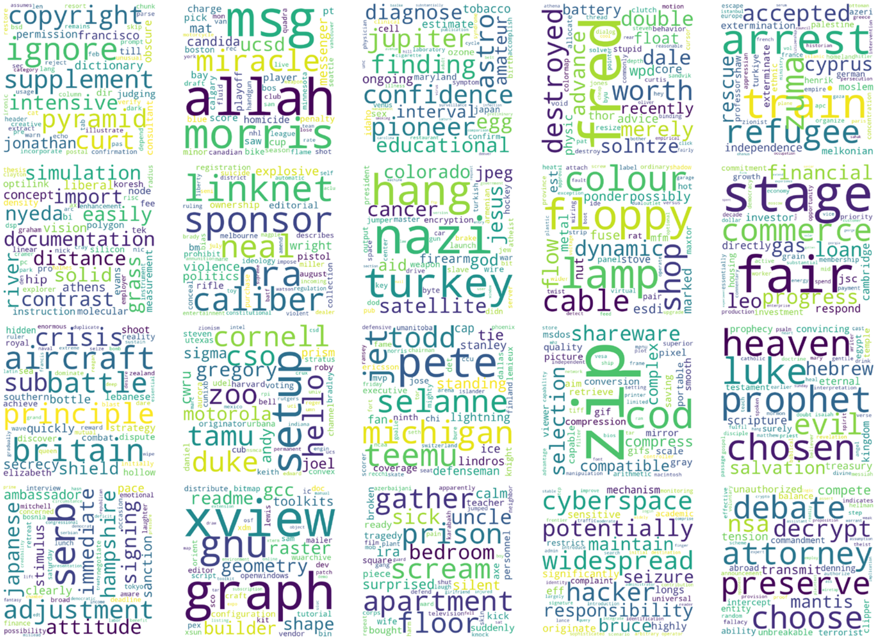

Fig. 7 Illustration of derived word-clusters by BROCCOLI NMF on the 20 Newsgroups dataset. The size of a word reflects its weight in the corresponding cluster $(X_{\cdot s})$

$X_{is} \geq 0$. In turn, the binary word-indicators of BROCCOLI RND and BROCCOLI NMF are visualized such that the size of a word in the cloud is proportional to the inverse of the number of clusters the word is assigned to. That is, those words which are unique to the respective cluster are larger than those words which are assigned to multiple clusters.

Looking at the visualizations of clusters, we see that the word *max* pops up prominently in many clusters. The word *max* obtains comparably very high term frequencies. The average term frequency of a word is equal to 1.59, and 99% of all words have a term frequency smaller than or equal to 8. The word *max* occurs in 149 posts and obtains term frequencies in $[1, 800]$. Hence, the word *max* attains exceptionally high term frequencies in a few posts and exhibits therewith a special role. The unusual high term frequencies of this word are handled differently among the clustering methods. While NN and OO give a high weight to this word in almost all clusters, BROCCOLI RND and BROCCOLI NMF reflect more general clustering structures. It is noteworthy that, as we can observe in Table 4, BROCCOLI RND and BROCCOLI NMF obtain a notably higher (worse) MSE% than competitors. The unusually high frequency of the word *max* vastly increases the approximation error for any cluster model that does not adapt to (and possibly overfit on) these particularly high word occurrences. Nevertheless, the approximation error exhibited by the model based on true class labels (i.e., NMF given Y) is comparable to that achieved by BROCCOLI RND and BROCCOLI NMF.

In any case, we can detect meaningful clusters which represent a specific topic for all clustering methods. Comparing the topics, we can see that BROCCOLI NMF provides a distinctive view on the dataset, identifying, for example, a *religion* cluster, which is not featured by the methods NN and OO. Hence, although BROCCOLI's optimization makes use of a relaxed objective, its results still provide another view on the data with respect to that provided by the relaxed counterparts NN and OO.

5.4.2 The distribution of observation-clusters over 20News categories

The differences between the cluster models are evident also when we look at the visual representation of the cluster overlaps with the *20News* categories, depicted in Fig. 8. In this figure, on the horizontal axis we show the *20News* categories at the bottom and super-categories at the top, while on the vertical axis we list the identified clusters. The intensity of each pixel reflects the normalized count of posts in the corresponding cluster and category. The cluster-category count is normalized such that every cluster has a maximum agreement of 1.0 with at least one of the categories, while the remaining agreement values range between 0.0 and 1.0. Consequently, we see at least one pixel per row having the highest intensity. The presence of multiple pixels per row with a high intensity means that such a cluster covers multiple categories with equally high agreement.

In Fig. 8, we can observe that all methods except BROCCOLI NMF yield clusters which overlap mostly with only one of the categories. For OO and NN, such a category is *comp.os.ms-windows*, while for BROCCOLI RND such a category is *comp.windows.x*. It is interesting to note that *comp.os.ms-windows* and *comp.windows.x* are the categories which contain few posts with an exceptionally

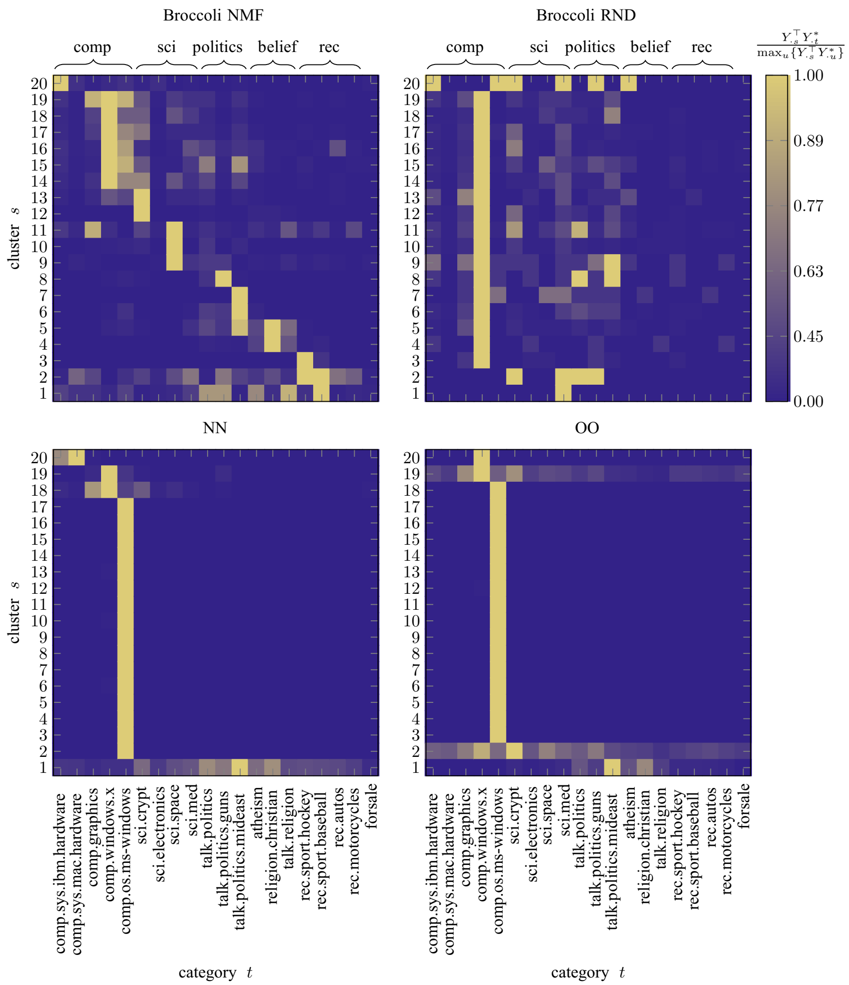

Fig. 8 Visualization of the overlap of Broccoli NMF clusters on the 20News dataset and the given categories

high frequency of the word *max*. From an optimization perspective this is not surprising, since the fuzzy clustering of NN and OO is able to notably decrease the MSE%, by specifically focusing on those data matrix entries with very high values, which strongly influence the (squared) approximation error. On the other hand, from a clustering perspective, this behavior is not necessarily desirable, since it can result in a very *one-sided* representation of the data, as shown in Fig. 8: the clusters are overfitting 12 posts containing the word *max* at least 45 times. As regards Broccoli RND, the imposed binary constraints result in slightly more widespread clusters, since they ensure that all features in one bicluster are approximated with the same value. However, Broccoli RND's clusters concentrate around one of the categories with the unusual high word occurrences. Here, the random initialization does not seem to

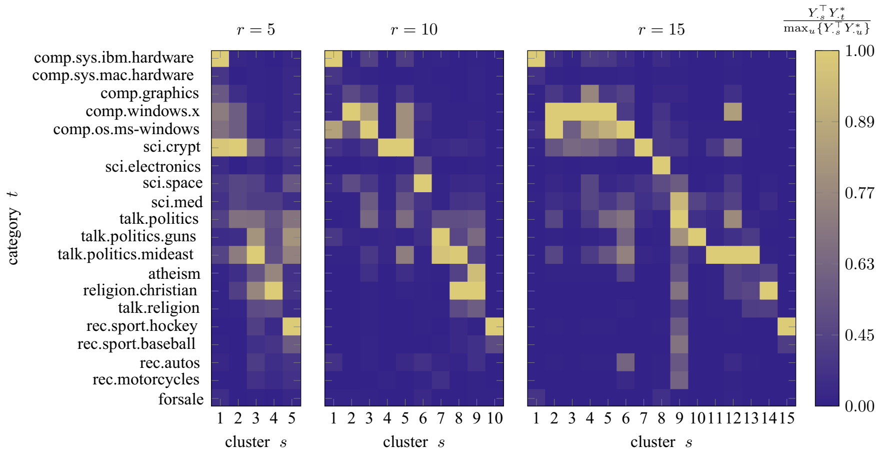

Fig. 9 Visualization of the overlap of BROCCOLI NMF clusters on the *20News* dataset for different ranks

be suitable to find a good starting point for the optimization: neither the performance measures nor the approximation error nor the visual analysis indicate that BROCCOLI RND reflect meaningful topic similarities.

In contrast, the clusters identified by BROCCOLI NMF are diverse and cover various categories. In particular, we can see clusters which clearly originate only from one category (e.g., *talk.politics.guns* in cluster 8), and others which summarize various topics (e.g., *comp* in clusters 14–19). Few clusters cover categories from different super-categories. For example, cluster 11 comprises posts from *comp.graphics* and *sci.space*, while cluster 5 models posts from *talk.politics.mideast* and *religion.christian*. However, even if they come from different super-categories, such specific combinations semantically make sense, and emphasize the ability of BROCCOLI NMF of catching such particular inter-topic relationships.

5.4.3 Influence of the rank

Besides the optimization aspects, there is one open problem for (bi)clustering applications in practice: the determination of the rank, i.e., of the most appropriate number of clusters. Although specifically addressing such a vast topic is out of the scope of this paper, in this subsection we qualitatively observe the influence of the rank on the results obtained on the *20News* dataset. In Table 4, we have already observed that BROCCOLI NMF identifies for some datasets a number of biclusters that is smaller than the specified rank. In Fig. 9 we show the category distribution of BROCCOLI NMF clusters, when we specify lower ranks $r \in \{5, 10, 15\}$. We observe that BROCCOLI NMF tends to model more general clusters for smaller ranks. For $r = 5$, there are two clusters focused on *sci.crypt*, which go either in the direction of *politics* or *comp*-related topics, a cluster representing the super-category *belief*, a cluster encompassing *politics* and *belief*, and a cluster encompassing *sports* and *politics*. As expected, the more we increase the rank, the more specific the clusters become, until they only cover few categories. It is noteworthy that, even when choosing a rank which apparently seems too low, BROCCOLI NMF still groups together categories that are somehow related, or belonging to the same super-category. This behavior is evident starting from $r = 10$, where sub-categories related to religion, talks and computers, respectively, appear to be reasonably grouped into distinct clusters.

5.5 Discussion on possible limitations

Our proposed penalized optimization approach is flexible and has the potential to become a novel general approach for the optimization of nonexclusive clustering structures based on matrix factorization. Notably, most of the popular clustering methods are based on (or can be viewed as) a matrix factorization with binary constraints, including k-means, spectral clustering, and variants of deep clustering. Despite these characteristics, in the following we highlight possible limitations of the proposed approach. We should first note that the adopted squared Frobenius norm is sensitive to data matrix entries having a magnitude that is much higher than the average. In other words, if there are very few data entries with very high values, then any factorization with a low MSE needs to fit the clustering to the unusual high-valued entries. This phenomenon is in contrast with the general concept of clustering, which is to group reoccurring patterns. We actually considered two examples of datasets which have few unusual high values: *birds* and *20News*. For these datasets, we observe a big gap between the 99.9th percentile and the maximum data value (cf. Table 3). For both datasets, BROCCOLI NMF is not able to achieve a low MSE and the performance measures give contrasting signals. This does not mean that the clusters identified by BROCCOLI NMF are not meaningful, as confirmed in the qualitative exploration we performed for *20News*. However, outliers with extremely high values aggravates the optimization.

This leads to the second limitation of our approach: the initialization. We have been able to derive a suitable heuristic for the initialization, which works well for most of the considered cases. However, other contexts might benefit from a different initialization. In practice, variations of the proposed initialization should be considered in future experiments, by varying, for example, the percentage of cluster indicators which are scaled to one.

Finally, there is the issue of selecting the rank. This is not a point in question specifically for the proposed method, but is related to a general open problem in clustering. There is always the possibility to apply popular heuristics such as the elbow method, which is based on the identification of an elbow in the curve of the approximation error, plotted against the rank. Moreover, we have also observed that, in some datasets, increasing the user-specified rank does not necessarily correspond to a higher number of clusters identified by BROCCOLI NMF. This means that such heuristics might be adopted to determine an upper bound of the clusters BROCCOLI should identify, but, in general, the determination of the rank based on theoretically justified grounds is still an open problem.

6 Conclusions

We have proposed the biclustering method Broccoli, which employs recent advances in optimization theory for the optimization of nonnegative tri-factorizations with binary constraints. To the best of authors' knowledge, Broccoli can be considered the first algorithm that is able to model the possible overlap between clusters, as well as the presence of outliers in the data, without requiring the user to specify characteristics about the obtained clustering (such as the amount of overlap or outliers), while returning definite, non-fuzzy cluster assignments. Employing the well-founded theory of proximal stochastic optimization, our method is guaranteed to converge to a local minimum in expectation (Driggs et al. 2020). Our method is based on the penalization of non-binary terms in the cluster-assignment matrices. The regularization weight of the penalizing terms is automatically updated during training, while the user has only to specify the step-size of these updates, which should be as small as possible in practice.

Our experiments on synthetic datasets show that our method is able to detect the underlying clustering structure and that it is robust to noise. Figure 3 shows that the direct optimization for binary cluster indicator matrices of Broccoli generally achieves higher performance measures (cf. Sect. 5.1) than the methods which compute a relaxed factorization, which are binarized according to the (generally unknown) ground truth. Broccoli relies on an initialization based on shortly optimized nonnegative matrix factorization. While this initialization method did not make a very notable difference for the synthetic data, which have a clear biclustering structure as ground truth, we have seen that the initialization becomes important for real-world data. The experiments on real-world data (cf. Table 4) show that the initialization of Broccoli with a shortly optimized nonnegative matrix factorization (Broccoli NMF) is suitable to find minima which attain a remarkably low approximation error, sometimes even lower than the one of relaxed factorizations. Finally, in our qualitative evaluation, we have observed that Broccoli NMF is able to derive meaningful clusters in terms of the found feature-clusters (cf. Figs. 4, 5, 6, 7.) and the observation clusters (cf. Fig. 8). Furthermore, we have briefly discussed the effect of the setting of the rank for Broccoli NMF (cf. Fig. 9). We found that for smaller ranks, Broccoli NMF tends to return more encompassing clusters, which overarch multiple categories and super-categories.

We can conclude that Broccoli allows swift adaptations of advances in the theory of nonconvex optimization. In addition, techniques to cope with specific data characteristics in matrix factorization can easily be transferred to the optimization scheme adopted in Broccoli. For example, a common technique for handling missing data in matrix factorization consists in the minimization of the approximation error only for the observed data matrix entries, i.e., on the non-missing data. This approach can be integrated in Broccoli by setting the partial derivatives in the gradient descent updates to zero in correspondence with the missing data indices.

This makes Broccoli a theoretically founded, practically well-performing and flexible approach which has the potential to spark further research on the optimization of non-exclusive clusterings in particular, and on the learning of discrete structures in general.

Acknowledgements Gianvito Pio acknowledges the support of Ministry of Universities and Research (MUR) through the project “Big Data Analytics”, AIM 1852414, activity 1, line 1.

Funding Open access funding provided by University degli Studi di Bari Aldo Moro within the CRUI-CARE Agreement.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Appendix A: Lipschitz constants of the batch-gradient

We briefly denote the Lipschitz constants, which define the step-sizes in Algorithm 1. Given the mean squared error function:

$$\mathrm{MSE}(C, X, Y) = \frac{1}{nm}\|D - YCX^\top\|^2,$$

the gradients with respect to single matrices are given by:

$$\begin{aligned}
\nabla_X \mathrm{MSE}(C, X, Y) &= \frac{2}{nm}(D - YCX^\top)^\top YC \\
\nabla_Y \mathrm{MSE}(C, X, Y) &= \frac{2}{nm}(D - YCX^\top)XC^\top \\
\nabla_C \mathrm{MSE}(C, X, Y) &= \frac{2}{nm}Y^\top(D - YCX^\top)X.
\end{aligned}$$

If we compute the gradients on a batch $\mathcal{M} \times \mathcal{I}$ or $\mathcal{J} \times \mathcal{N}$, for $\mathcal{J} \subseteq \mathcal{M}$ and $\mathcal{I} \subseteq \mathcal{N}$, and $\mathcal{N} = \{1, \dots, n\}$ and $\mathcal{M} = \{1, \dots, m\}$, then we obtain the following batch gradients:

$$\begin{aligned}
\nabla_X \mathrm{MSE}(C, X, Y) \restriction_{\mathcal{J}\times\mathcal{N}} &= \frac{2}{|\mathcal{J}|\,n}(D_{\mathcal{J}\cdot} - Y_{\mathcal{J}\cdot}CX^\top)^\top Y_{\mathcal{J}\cdot}C \\
\nabla_Y \mathrm{MSE}(C, X, Y) \restriction_{\mathcal{M}\times\mathcal{I}} &= \frac{2}{|\mathcal{I}|\,m}(D_{\cdot\mathcal{I}} - YCX_{\mathcal{I}\cdot}^\top)X_{\mathcal{I}\cdot}C^\top \\
\nabla_C \mathrm{MSE}(C, X, Y) \restriction_{\mathcal{J}\times\mathcal{N}} &= \frac{2}{|\mathcal{J}|\,n}Y_{\mathcal{J}\cdot}^\top(D_{\mathcal{J}\cdot} - Y_{\mathcal{J}\cdot}CX^\top)X \\
\nabla_C \mathrm{MSE}(C, X, Y) \restriction_{\mathcal{M}\times\mathcal{I}} &= \frac{2}{|\mathcal{I}|\,m}Y^\top(D_{\cdot\mathcal{I}} - YCX_{\mathcal{I}\cdot}^\top)X_{\mathcal{I}\cdot}.
\end{aligned}$$

The Lipschitz constants of the batch gradients are then given by:

$$
\begin{aligned}
L_{\nabla_X \upharpoonright_{\mathcal{J}\times\mathcal{N}}} &= \frac{2}{|\mathcal{J}|\,n}\|(Y_{\mathcal{J}\cdot}C)^\top Y_{\mathcal{J}\cdot}C\| \\
L_{\nabla_Y \upharpoonright_{\mathcal{M}\times\mathcal{I}}} &= \frac{2}{|\mathcal{I}|\,m}\|(X_{\mathcal{I}\cdot}C^\top)^\top X_{\mathcal{I}\cdot}C^\top\| \\
L_{\nabla_C \upharpoonright_{\mathcal{J}\times\mathcal{N}}} &= \frac{2}{|\mathcal{J}|\,n}\|Y_{\mathcal{J}\cdot}^\top Y_{\mathcal{J}\cdot}\|\,\|X^\top X\|. \\
L_{\nabla_C \upharpoonright_{\mathcal{M}\times\mathcal{I}}} &= \frac{2}{|\mathcal{I}|\,m}\|Y^\top Y\|\,\|X_{\mathcal{I}\cdot}^\top X_{\mathcal{I}\cdot}\|.
\end{aligned}
$$

Appendix B: Discussion on clustering agreement measures

Rabbany and Zaïane (2015) propose four main agreement measures for overlapping clusters. In the following, we first discuss the two measures which we omitted from our experimental evaluation, namely:

$$
\begin{aligned}
I_{\mathrm{ARI}}(Y, Y^*) &= \frac{\|Y^\top Y^{*\top}\|^2 - \frac{1}{m^2}|YY^\top||Y^*Y^{*\top}|}{\|YY^\top\|^2 + \|Y^*Y^{*\top}\|^2 - \frac{1}{m^2}|YY^\top||Y^*Y^{*\top}|}, \\
I_F(Y, Y^*) &= \frac{\|Y^\top Y^{*\top}\|^2}{\|YY^\top\|^2 + \|Y^*Y^{*\top}\|^2}.
\end{aligned}
$$

The I_{ARI}-measure is an extension of the Adjusted Randomized Index (ARI) for overlapping clusters. We observe that I_{ARI} is related to I_F. The difference is that I_{ARI} subtracts a term from both, the nominator and the denominator of the I_F measure. The subtracted term is introduced by the normalization of the ARI measure, which adjusts for random cluster correspondences by assuming independence of the matrices $YY^\top$ and $Y^*Y^{*\top}$. Both measures are not scaling invariant. That is, the condition $I(\alpha Y, Y^*) = I(Y, Y^*)$ does generally not hold for $\alpha \in \mathbb{R}$. As a result, the above mentioned measures are sensitive to the norm of the cluster indicator matrices. This does not have to be an issue, but we noticed that the scaling sensitivity leads to inaccurate reflections of the clustering performance. In a preliminary experimental evaluation, we found out that clusters consisting of a mix of observations coming from multiple ground truth clusters (e.g., 20% observations randomly taken from 6 different ground truth clusters) obtain a much higher (often up to 10 times higher) I_{ARI} and I_F measurement than clusterings which merge two or three whole ground truth clusters. Since the possible reward which is given to clusterings picking few observations from various ground truth clusters (thus scattering the ground truth structure) provides misleading performance indications, we do not consider I_{ARI} and I_F in our experimental evaluation.

The two other clustering measures proposed by Rabbany and Zaïane (2015) are the following ones:

$$I_{\cos}(Y, Y^*) = \frac{\|Y^\top Y^{*\top}\|^2}{\|YY^\top\|\|Y^*Y^{*\top}\|},$$
$$I_{\text{sub}}(Y, Y^*) = \frac{\|Y^\top Y^{*\top}\|}{\|Y\|\|Y^*\|}.$$

We observe that these measures are scaling invariant, that is the condition $I(\alpha Y, Y^*) = I(Y, Y^*)$ holds for all $\alpha \in \mathbb{R}$. Indeed, the $I_{\cos}$ measure can be interpreted as the cosine of the angle of the vectorized matrices $\text{vec}(YY^\top)$ and $\text{vec}(Y^*Y^{*\top})$.

Proposition 1 *For matrices $Y, Y^* \in \mathbb{R}^{m\times r}$ we have*

$$I_{\cos}(Y, Y^*) = \cos(\angle(\text{vec}(YY^\top), \text{vec}(Y^*Y^{*\top}))),$$

where

$$\text{vec}(YY^\top) = \begin{pmatrix} Y_{1\cdot}Y_{1\cdot}^\top \\ \vdots \\ Y_{m\cdot}Y_{m\cdot}^\top \end{pmatrix}$$

Proof Given two cluster indicator matrices $Y \in \{0, 1\}^{m\times r}$ and $Y^* \in \{0, 1\}^{m\times r^*}$, the clustering agreement is according to the definition of the Frobenius norm via the trace equal to

$$\|Y^\top Y^*\|^2 = \text{tr}(Y^\top Y^* Y^{*\top} Y) = \text{tr}(Y^* Y^{*\top} YY^\top).$$

The trace defines an inner product, the Frobenius inner product, which can be defined for matrices $A, B \in \mathbb{R}^{a\times b}$ by means of the vec operator:

$$\langle A, B\rangle = \text{tr}(A^\top B) = \text{vec}(A)^\top \text{vec}(B).$$

Correspondingly, we can write the Frobenius norm by means of the vec-operator as

$$\|A\| = \sqrt{\langle A, A\rangle} = \sqrt{\text{vec}(A)^\top \text{vec}(A)} = \|\text{vec}(A)\|.$$

As a result, we obtain for the $I_{\cos}$-measure the following presentation:

$$\begin{aligned} I_{\cos}(Y, Y^*) &= \frac{\|Y^\top Y^*\|^2}{\|YY^\top\|\|Y^*Y^{*\top}\|} \\ &= \frac{\text{vec}(Y^*Y^{*\top})^\top \text{vec}(YY^\top)}{\|\text{vec}(YY^\top)\|\,\|\text{vec}(Y^*Y^{*\top})\|} \\ &= \cos(\angle(\text{vec}(YY^\top), \text{vec}(Y^*Y^{*\top}))). \end{aligned}$$

□

The $I_{\cos}$ measure indicates the similarity between two clusterings Y and Y^* as the cosine similarity of the agreement matrices $YY^\top$ and $Y^*Y^{*\top}$.

In comparison, the I_{sub} measure is more related to a weighted similarity measurement of the subspaces, spanned by the columns of Y and Y^*. Formally, let $Y = U\Sigma V^\top$ and $Y^* = U^*\Sigma^*V^{*\top}$ be the singular value decompositions of the cluster indicator matrices. Then we have

$$\begin{aligned} \|Y^\top Y^*\|^2 &= \|V\Sigma^\top U^\top U^*\Sigma^*V^{*\top}\|^2 \\ &= \|\Sigma^\top U^\top U^*\Sigma^*\|^2 \\ &= \sum_{s,t} \sigma_s^2 \sigma_t^{*2} (U_{\cdot s}^\top U_{\cdot t}^*)^2. \end{aligned}$$

The columns of the matrix $U \in \mathbb{R}^{m\times r}$ indicate an orthogonal basis of the subspace spanned by the columns of Y. The normalization term is equal to:

$$\|Y\|^2\|Y^*\|^2 = \|\Sigma\|^2\|\Sigma^*\|^2 = \sum_{s,t} \sigma_s^2\sigma_t^{*2}.$$

As a result, the I_{sub} measure returns a weighted comparison of the subspaces induced by the cluster indicator columns, as follows:

$$I_{\text{sub}}(Y, Y^*) = \frac{\sum_{s,t} \sigma_s^2\sigma_t^2 (U_{\cdot s}^\top U_{\cdot t}^*)}{\sum_{s,t} \sigma_s^2\sigma_t^2}.$$

References

Asteris M, Papailiopoulos D, Dimakis AG (2015) Orthogonal NMF through subspace exploration. In: Advances in neural information processing systems, pp 343–351

Barracchia EP, Pio G, D'Elia D, Ceci M (2020) Prediction of new associations between NCRNAS and diseases exploiting multi-type hierarchical clustering. BMC Bioinform 21(1):70

Bauckhage C (2015) K-means clustering is matrix factorization. arXiv preprint arXiv:1512.07548

Bolte J, Sabach S, Teboulle M (2014) Proximal alternating linearized minimization or nonconvex and nonsmooth problems. Math Program 146(1–2):459–494

Boutell MR, Luo J, Shen X, Brown CM (2004) Learning multi-label scene classification. Pattern Recogn 37(9):1757–1771

Briggs F, Huang Y, Raich R, Eftaxias K, Lei Z, Cukierski W, Hadley SF, Hadley A, Betts M, Fern XZ et al (2013) New methods for acoustic classification of multiple simultaneous bird species in a noisy environment. In: 2013 IEEE international workshop on machine learning for signal processing (MLSP), pp 1–8

Cai D, He X, Han J, Huang TS (2011) Graph regularized nonnegative matrix factorization for data representation. IEEE Trans Pattern Anal Mach Intell 33(8):1548–1560

Cheng Y, Church GM (2000) Biclustering of expression data. In: Proceedings of the eighth international conference on intelligent systems for molecular biology, vol 8, pp 93–103

Cho H, Dhillon IS, Guan Y, Sra S (2004) Minimum sum-squared residue co-clustering of gene expression data. In: Proceedings of the SIAM international conference on data mining (SDM), pp 114–125

Del Buono N, Pio G (2015) Non-negative matrix tri-factorization for co-clustering: an analysis of the block matrix. Inf Sci 301:13–26

Dhillon IS (2001) Co-clustering documents and words using bipartite spectral graph partitioning. In: Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining (KDD), pp 269–274

Ding C, Li T, Peng W (2006a) Nonnegative matrix factorization and probabilistic latent semantic indexing: equivalence chi-square statistic, and a hybrid method. AAAI 42:137–143

Ding C, Li T, Peng W, Park H (2006b) Orthogonal nonnegative matrix t-factorizations for clustering. In: Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining (KDD), pp 126–135

Diplaris S, Tsoumakas G, Mitkas PA, Vlahavas I (2005) Protein classification with multiple algorithms. In: Panhellenic conference on informatics, pp 448–456

Driggs D, Tang J, Davies M, Schönlieb CB (2020) Spring: a fast stochastic proximal alternating method for non-smooth non-convex optimization. arXiv preprint arXiv:2002.12266

Elisseeff A, Weston J (2002) A kernel method for multi-labelled classification. In: Advances in neural information processing systems, pp 681–687

Gaul W, Schader M (1996) A new algorithm for two-mode clustering. In: Data analysis and information systems. Springer, pp 15–23

Han J, Song K, Nie F, Li X (2017) Bilateral k-means algorithm for fast co-clustering. In: AAAI, pp 1969–1975

Hardt M, Recht B, Singer Y (2016) Train faster, generalize better: stability of stochastic gradient descent. In: Proceedings of the international conference on machine learning (ICML), pp 1225–1234

Hartigan JA (1972) Direct clustering of a data matrix. J Am Stat Assoc 67(337):123–129

Hess S, Morik K, Piatkowski N (2017) The PRIMPING routine—tiling through proximal alternating linearized minimization. Data Min Knowl Discovery (DAMI) 31(4):1090–1131

Hochreiter S, Bodenhofer U, Heusel M, Mayr A, Mitterecker A, Kasim A, Khamiakova T, Van Sanden S, Lin D, Talloen W, Bijnens L, Göhlmann H, Shkedy Z, Clevert DA (2010) Fabia: factor analysis for bicluster acquisition. Bioinformatics (Oxford, England) 26:1520–7

Hoffer E, Hubara I, Soudry D (2017) Train longer, generalize better: closing the generalization gap in large batch training of neural networks. In: Advances in neural information processing systems (NIPS), pp 1731–1741

Kluger Y (2003) Spectral biclustering of microarray data: coclustering genes and conditions. Genome Res 13(4):703–716

Koyutürk M, Grama A (2003) PROXIMUS: a framework for analyzing very high dimensional discrete-attributed datasets. In: Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining (KDD), pp 147–156

Laclau C, Brault V (2019) Noise-free latent block model for high dimensional data. Data Min Knowl Discovery (DAMI) 33(2):446–473

Li T (2005) A general model for clustering binary data. In: Proceedings of the ACM SIGKDD international conference on knowledge discovery in data mining (KDD), pp 188–197

Lloyd S (1982) Least squares quantization in PCM. IEEE Trans Inf Theory 28(2):129–137

Long B, Zhang ZM, Yu PS (2005) Co-clustering by block value decomposition, vol '05. Association for Computing Machinery, New York, NY, USA, KDD, pp 635–640

Mirkin B, Arabie P, Hubert LJ (1995) Additive two-mode clustering: the error-variance approach revisited. J Classif 12(2):243–263

Nie F, Wang X, Deng C, Huang H (2017) Learning a structured optimal bipartite graph for co-clustering. In: Advances in neural information processing systems (NIPS), pp 4129–4138

Parikh N, Boyd S et al (2014) Proximal algorithms. Found Trends Optim 1(3):127–239

Pio G, Ceci M, Loglisci C, D'Elia D, Malerba D (2012) Hierarchical and overlapping co-clustering of MRNA: MIRNA interactions. In: ECAI 2012, IOS Press, frontiers in artificial intelligence and applications, vol 242, pp 654–659

Pio G, Ceci M, D'Elia D, Loglisci C, Malerba D (2013) A novel biclustering algorithm for the discovery of meaningful biological correlations between micrornas and their target genes. BMC Bioinform 14(S–7):S8

Pio G, Ceci M, Malerba D, D'Elia D (2015) Comirnet: a web-based system for the analysis of MIRNA-gene regulatory networks. BMC Bioinform 16(S–9):S7

Pompili F, Gillis N, Absil PA, Glineur F (2014) Two algorithms for orthogonal nonnegative matrix factorization with application to clustering. Neurocomputing 141:15–25

Rabbany R, Zaïane OR (2015) Generalization of clustering agreements and distances for overlapping clusters and network communities. Data Min Knowl Disc 29(5):1458–1485

Song K, Yao X, Nie F, Li X, Xu M (2020) Weighted bilateral k-means algorithm for fast co-clustering and fast spectral clustering. Pattern Recognit 109:107560

Trohidis K, Tsoumakas G, Kalliris G, Vlahavas IP (2008) Multi-label classification of music into emotions. ISMIR 8:325–330

Vichi M (2001) Double k-means clustering for simultaneous classification of objects and variables. In: Advances in classification and data analysis, pp 43–52

Wang H, Nie F, Huang H, Makedon F (2011) Fast nonnegative matrix tri-factorization for large-scale data co-clustering. In: Proceedings of the international joint conference on artificial intelligence (IJCAI), p 1553

Wang J, Tian F, Yu H, Liu CH, Zhan K, Wang X (2018) Diverse non-negative matrix factorization for multiview data representation. IEEE Trans. Cybern. 48(9):2620–2632

Whang JJ, Dhillon IS (2017) Non-exhaustive, overlapping co-clustering. In: Proceedings of the ACM conference on information and knowledge management (CIKM), pp 2367–2370

Yang J, Wang H, Wang W, Yu P (2005) An improved biclustering method for analyzing gene expression profiles. Int J Artif Intell Tools 14:771–790

Yokota T, Kawai K, Sakata M, Kimura Y, Hontani H (2019) Dynamic pet image reconstruction using nonnegative matrix factorization incorporated with deep image prior. In: Proceedings of the IEEE/CVF international conference on computer vision (ICCV)

Yoo J, Choi S (2010) Orthogonal nonnegative matrix tri-factorization for co-clustering: multiplicative updates on Stiefel manifolds. Inf Process Manag 46(5):559–570

Zha H, He X, Ding C, Simon H, Gu M (2001) Bipartite graph partitioning and data clustering. In: Proceedings of the international conference on information and knowledge management, pp 25–32

Zhang Z, Li T, Ding C, Zhang X (2007) Binary matrix factorization with applications. In: IEEE International conference on data mining (ICDM), pp 391–400

Zhang ZY, Li T, Ding C, Ren XW, Zhang XS (2010) Binary matrix factorization for analyzing gene expression data. Data Min. Knowl. Discov (DAMI) 20(1):28

Zhang ZY, Wang Y, Ahn YY (2013) Overlapping community detection in complex networks using symmetric binary matrix factorization. Phys Rev E 87(6):062803

Zhou J, Qi J (2011) Fast iterative image reconstruction using sparse matrix factorization with GPU acceleration. In: Progress in biomedical optics and imaging—proceedings of SPIE 7961

Zhou X, Leonardos S, Hu X, Daniilidis K (2015) 3d shape estimation from 2d landmarks: A convex relaxation approach. In: proceedings of the IEEE conference on computer vision and pattern recognition, pp 4447–4455