Expert Systems With Applications 191 (2022) 116219

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A multi-block clustering algorithm for high dimensional binarized sparse data

Zsolt T. Kosztyán [a,b,c,*], András Telcs [a,b,d], János Abonyi [e]

[a] University of Pannonia, Department of Quantitative Methods, Egyetem str. 10., H-8200, Veszprém, Hungary
[b] MTA-PE Budapest Ranking Research Group, Piarista str. 4., H-1052 Budapest, Hungary
[c] Institute of Advanced Studies, Kőszeg (iASK), Charnel str. 10., H-9730 Kőszeg, Hungary
[d] Department for Computational Sciences, Wigner Research Centre for Physics,, H-1121 Budapest, Hungary
[e] MTA-PE Lendület Complex Systems Monitoring Research Group, University of Pannonia, Egyetem str. 10., H-8200 Veszprém, Hungary

ARTICLE INFO

Keywords:
Multidimensional clustering
High dimensional data
Ranking
Higher educational institutes

ABSTRACT

We introduce a multidimensional multiblock clustering (MDMBC) algorithm in this paper. MDMBC can generate overlapping clusters with similar values along clusters of dimensions. The parsimonious binary vector representation of multidimensional clusters lends itself to the application of efficient meta-heuristic optimization algorithms. In this paper, a hill-climbing (HC) greedy search algorithm has been presented that can be extended by several stochastic and population-based meta-heuristic frameworks. The benefits of the algorithm are demonstrated in a bi-clustering benchmark problem and in the analysis of the Leiden higher education ranking system, which measures the scientific performance of 903 institutions along four dimensions of 20 indicators representing publication output and collaboration in different scientific fields and time periods.

1. Introduction

Data-driven scientific, economical and technological research and development increasingly necessitate the use of efficient high-dimensional data-mining methods, especially those that can find homogeneous subsets of data in more dimensions. Two-(Martínez, Morán, & Peña, 2006) and multi-step (Amato et al., 2006), bi- (Cheng & Church, 2000) and tri- clustering (Ignatov, Gnatyshak, Kuznetsov, & Mirkin, 2015) algorithms are increasingly popular methods in data mining. The rough set concept has been also utilized (Michalak & Ślezak, 2018; Wang, Miao, Li, & Zhang, 2007) which forms a promising group of bi-clustering algorithms.

The main difference between two- and multistep clustering and bi-, tri- or coclustering is that in the latter case, dimensional selections run simultaneously (Strauch et al., 2007). Although bi- and tri-clustering algorithms are mostly helpful for bioinformatics-relevant applications (Swathypriyadharsini & Premalatha, 2018), thanks to the effectiveness of these methods in finding complex homogeneous groups of objects (see, e.g., Kaiser & Leisch, 2008; Pontes, Giráldez, & Aguilar-Ruiz, 2015), bi- and tri-clustering are becoming utilized more widely, such as in the fields of business studies (Dolnicar, Kaiser, Lazarevski, & Leisch, 2012), knowledge management (Kaytoue et al., 2015), and in more general infocube-based applications (Lamani, Erraha, Elkyal, & Sair, 2019).

The mined homogeneous submatrices are also called *blocks* (Govaert & Nadif, 2008). The infocubes of online analytical processing (OLAP) (Jain, Taygi, Sharma, & Khatri, 2019) and block-(Hatano, Fukunaga, Maehara, & Kawarabayashi, 2017; Oktar & Turkan, 2018; Vinayak, Hassibi, & EDU, 2016) and coclustering (Forero, Baxley, & Capella, 2019) algorithms aim to find homogeneous disjoint blocks, while multiclustering algorithms aim to find disjoint clusters in a multidimensional dataset (Hu & Pei, 2018; Wang et al., 2018). Overlaps between blocks are permitted in non-exclusive bi- and tri-clustering (Pontes et al., 2015). As our best knowledge, there is no clustering algorithm that can find clusters in high dimensional data. Thus, it is the aim of this paper to fill this gap and to develop a robust and easily scalable method for clustering high dimensional binary or binarized data.

Permitting overlaps is important because it allows biclusters to share their conditions and can provide additional information (Kosztyán, Banász, Csányi, & Telcs, 2019) and ensure better performance, e.g., better image segmentation (Rahaman et al., 2019). The interpretability of the results deepens on the number and the homogeneity of the clusters. Allowing cluster overlaps helps in finding a small number of large clusters, so we are interested in an algorithm that allows cluster overlaps and can be tuned to handle the trade-off of cluster size and homogeneity.

[*] Corresponding author at: University of Pannonia, Department of Quantitative Methods, Egyetem str. 10., H-8200, Veszprém, Hungary.
E-mail addresses: kzst@gtk.uni-pannon.hu (Z.T. Kosztyán), telcs.andras@wigner.mta.hu (A. Telcs), janos@abonyilab.com (J. Abonyi).

https://doi.org/10.1016/j.eswa.2021.116219
Received 12 December 2020; Received in revised form 25 June 2021; Accepted 9 November 2021
Available online 20 November 2021

Bi-clustering, and therefore also the tri- and higher-dimensional clustering problems, is an NP-hard problem (Tanay, Sharan, & Shamir, 2002), and it is much more complex than the classical clustering tasks (Divina & Aguilar-Ruiz, 2006), so most of the methods for solving it are based on heuristic (Michalak, 2012; Wang et al., 2007) or meta-heuristic algorithms (Pontes et al., 2015). The development of an effective heuristic as well as the use of a suitable cost function for guiding the search are critical factors for finding significant bi-clusters (Pontes et al., 2015) and tri-clusters (Henriques & Madeira, 2018). Nevertheless, in higher dimensions, parallel computation may accelerate the search. Many bi-clustering approaches have been proposed based on evolutionary algorithms, such as genetic algorithms (GA) (Cheng & Church, 2000; Gusenleitner, Howe, Bentink, Quackenbush, & Culhane, 2012), pattern search (PS) (Pandey, Atluri, Steinbach, Myers, & Kumar, 2009), ant colony optimization (ACO) (Liu, Li, Hu, & Chen, 2009), and swarm intelligence (SI) (Veroneze, de França, & Zuben, 2011) and heuristic, such as rough set clustering (Michalak & Ślezak, 2018; Michalak & Stawarz, 2013). The GA is one of the oldest nature-inspired meta-heuristics, and this approach has been broadly applied to solve problems in many fields of engineering and science. One of the main advantages of this method is that it can be easily parallelized (Orzechowski, Sipper, Huang, & Moore, 2018). Even in the case of a serial GA, a larger subset of the whole space of solutions is explored, and at the same time, the algorithm avoids becoming trapped in a local optimum. For these reasons, population-based and meta-heuristic algorithms are well suited to the bi-clustering problem. In addition, in the case of parallel computation, several sub-populations will be stimulated to explore distinct regions of the search space (Orzechowski et al., 2018).

Although classical genetic algorithms and simulated annealing are powerful in the exploration of the search space and exploitation of the extracted information during the search, the performance of these algorithms can be significantly improved by utilizing problem-relevant representations and search operations. The main contribution of our work is that the multidimensional clusters are represented by a parsimonious binary vector that lends itself to the application of efficient meta-heuristic optimization algorithms.

This work aims to develop an elementary building block of a family of multidimensional clustering algorithms. Greedy hill-climbing (HC) algorithm is a very standard procedure in meta-heuristic-based clustering techniques, e.g. such strategy is used in the widely applied DBSCAN algorithm (Chandana, Srinivas, & Kumar, 2014; Matioli, Santos, Kleina, & Leite, 2018) and the approach has also been proven applicable in bi-clustering (Ayadi, Elloumi, & Hao, 2010; Hu et al., 2014).

According to these, the main contributions of this work are the following:

- We introduce a multidimensional multi-block clustering (MDMBC) algorithm that can generate overlapping clusters with similar values along clusters of dimensions. The cost function of the proposed algorithm can be easily modified according to how the homogeneity of the clusters are measured and how the resulted clusters will be utilized and interpreted (see Section 2.1).
- We propose generalized parsimonious binary vector-based representation of bi- and tri-clustering problems that can be generalized to a higher-dimensions that creates the opportunity to integrate with most of the heuristic optimization methods, such as genetic algorithms or simulated annealing (see Section 2.2).
- We present a greedy strategy for growing the clusters. The proposed greedy search algorithm can be considered a hill-climbing optimization algorithm. The algorithm has similar performance to the widely applied iBBiG; however, it has some benefits, as it finds clusters (blocks) that are maximal in every possible direction of the selected features, and by controlling the required block purity and the minimal size of the blocks, users can fully control what kinds of blocks are identified from the data. The presented core algorithm can be incorporated into any meta-heuristic or population-based search framework, so the paper proposes a family of multidimensional multi-block clustering algorithms (see Section 2.3).
- As the algorithm can generate significantly overlapping clusters and the selection of the informative features is a complex optimization problem, we remove the redundant clusters that do not provide any additional information and select features based on their statistical validation (see Section 2.4).
- The benefits of the algorithm are demonstrated in a bi-clustering benchmark problem (Section 3.1), which serve as a proof of concept study/analysis of the described method. As the possible extensions of the algorithm are almost infinite, in this paper, we present the performance of the core algorithm with some important extensions that support maximal coverage of the dataset, controlling the minimum size, achieving homogeneity and statistical significance and addressing overlaps of the identified blocks. A high dimensional clustering problem generator has also been proposed to generate reproducible tests to demonstrate the sensitivity of the performance and runtime to the hyper-parameters of the proposed MDMBC algorithm.
- A detailed application study is presented through the analysis of the four-dimensional CWTS Leiden Ranking database, which measures the scientific performance of 903 institutions along four dimensions of 20 indicators representing publication output and collaboration in different scientific fields and time periods. Kosztyán et al. (2019) proposed a bi-clustering method to specify leagues and partial rankings of the higher education systems of countries based on this dataset; however, their method considered only two dimensions (indicators and countries) simultaneously, while most of the ranking indicators are available in several scientific fields and in several time periods; therefore, in this field, a higher-dimension block clustering method is required. The presented results illustrate how the variables of these additional dimensions influence the results of the clustering (see Section 3.2).
- Section 4 discusses the main properties of the algorithm and we summarize the results of the proposed algorithm and conveys our hope that this paper and the related code can motivate further research on finding clusters in multidimensional datacubes.

2. Multidimensional multiblock clustering algorithm

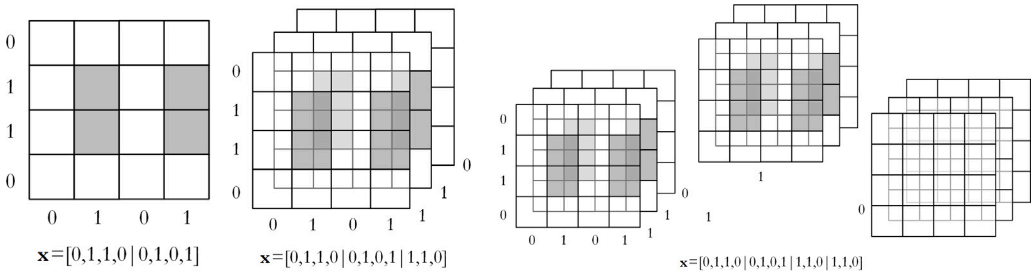
2.1. Problem formulation

In the studied n-dimensional data, sets of objects $I_1 = \{i_{1,1} \dots, i_{1,n_1}\}$ are characterized by sets of categorical variables $I_j = \{i_{j,1} \dots, i_{j,n_j}\}$, $j = 2, \dots, n$. To illustrate such a dataset, let us consider the studied four-dimensional datacube of the Leiden database, where the set of universities listed in the set I_1 are represented by the set I_2 of scientometric measures, such as $I_{2,1}$=number of publications and $I_{2,2}$=number of citations, in different scientific fields that define the third dimension, such as $I_3 = \{$mathematical and computer science, social sciences and humanities.$\}$ in different time periods, such as $I_4 = \{$2009–13,2014–17, $\dots\}$.

When the data are stored in an n-dimensional matrix $\mathbf{Z}$, the aim of clustering is to find homogeneous multidimensional blocks $\mathbf{Z}^c = g(\mathbf{Z}, C^c)$ $c = 1, \dots, K$ defined as n-ary Cartesian products of the subsets $C_j^c \subseteq I_j$ of these features:

$$C^c = C_1^c \times, \dots, \times C_n^c, \tag{1}$$

where the function $g(\mathbf{Z}, C^c)$ represents how the submatrix $\mathbf{Z}^c$ of $\mathbf{Z}$ is extracted based on the subsets of the features. When $C_j^k \cap C_j^l = \emptyset, \forall k, j \in 1, \dots, n_c$, every item is assigned to at most one cluster, which represents a special case of n-dimensional clustering that can be interpreted as *seriation* of $\mathbf{Z}$. When the features are not assigned exclusively, the


0
1
1
0
0 1 0 1
x=[0,1,1,0 | 0,1,0,1]
x=[0,1,1,0 | 0,1,0,1 | 1,1,0]
x=[0,1,1,0 | 0,1,0,1 | 1,1,0 | 1,1,0]

(a) 2D block (b) 3D block (c) 4D block

Fig. 1. Binary vector-based representation of n-dimensional blocks.

clusters can overlap, which allows flexibility in finding informative subsets of data.

Finding homogeneous groups in data is valuable because it can be used for segmentation of the objects by determining the subsets of features that characterize the clusters, e.g., finding an elite group of universities with high academic performance in the last three years in engineering and economics. The homogeneity of the clusters is measured by a cost function $H^c = f(\mathbf{Z}^c)$, such as standard deviation or entropy, that measures how homogeneous the subset $\mathbf{Z}^c$ of the n-dimensional data $\mathbf{Z}$ is. The clustering problem is formalized as the minimization of the sum of these H^c values:

$$\min_{C^1,\ldots,C^K} H = \sum_{c=1}^{K} H^c = \sum_{c=1}^{K} f\left(g\left(\mathbf{Z}, C^c\right)\right). \tag{2}$$

In classical clustering, a scalar centroid m^c represents the cluster prototype. This centroid can be defined by the mean or the median of the block $\mathbf{Z}^c = g(\mathbf{Z}, C^c)$, so in this case, the cost function $f()$ measures the homogeneity of the cluster as $H^c = \|m^c - g(\mathbf{Z}, C^c)\|$.

The cost function can be modified to minimize the overlaps between the clusters if the user would like to find such clusters. As the clusters act as a mixture of K block models used to approximate the $\mathbf{Z}$ multidimensional array, the cost function can be modified also to try to cover the multidimensional matrix as much as possible:

$$\min_{C^1,\ldots,C^K} H + \lambda E = \sum_{c=1}^{K} \|m^c - g(\mathbf{Z}, C^c)\| + \lambda \left\| \mathbf{Z} - \sum_{c=1}^{K} g(\mathbf{Z}, C^c) \right\|. \tag{3}$$

2.2. Binary representation of the search space

To design an optimization algorithm that is efficient in finding the clusters $\mathbf{Z}^c = g(\mathbf{Z}, C^c)$ $c = 1, \ldots, K$ as n-ary Cartesian products of the subsets $C_j^c \subseteq I_j$ of the features, there is a need for an efficient representation of the variables of the optimization problem. In the proposed algorithm, the subsets C_j^c are represented by binary vectors $\mathbf{x}_j^c = \left[x_{j,1}^c, \ldots, x_{j,n_j}^c\right]^T$, where the nonzero elements $x_{j,k}^c = 1$ represent $I_{j,k} \in C_j^c$, where $I_{j,k}$ is the kth element of the set I_j.

With this representation, the elements and blocks of the array $\mathbf{Z}$ are assigned (see Fig. 1); e.g., the $z_{5,3,2,1}$-th element of a four-dimensional array $\mathbf{Z}$ is associated with $x_{1,5} = 1$, $x_{2,3} = 1$, $x_{3,2} = 1$ and $x_{4,1} = 1$. With this representation, every cluster is represented by a set of vectors $\mathbf{x}^c = \left\{\mathbf{x}_1^c, \ldots, \mathbf{x}_n^c\right\}$. With this formalization, the clustering problem is formulated as a binary optimization problem defined by the cost function:

$$\min_{\mathbf{x}^1,\ldots,\mathbf{x}^K} H = \sum_{c=1}^{K} H^c = \sum_{c=1}^{K} \|m^c - g(\mathbf{Z}, \mathbf{x}^c)\| \tag{4}$$

and the following related constraints:

- In every dimension, a minimum number category $n_j^{min} \geq 1$ should be involved in a cluster

$$n_j^c = \sum_{k=1}^{n_j} x_{j,k}^c > n_j^{min}, \forall j, c \tag{5}$$

- The size of the cluster $N^c = \prod n_j^c$ should be larger than a threshold,

$$N^c = \prod_{j=1}^{n} \sum_{k=1}^{n_j} x_{j,k}^c > N^{min}, \forall c \tag{6}$$

- and the homogeneity ("purity") of the clusters should be higher than a threshold value,

$$H^c \geq tr. \tag{7}$$

2.3. Details of the clustering algorithm

The previously presented optimization problem can be solved by any meta-heuristic binary-valued optimization algorithm that can efficiently handle the complexity of the problem. Although classical algorithms, such as genetic algorithms and simulated annealing, are powerful in the exploration of the search space and exploitation of information extracted during the search, the performance of these algorithms can be significantly improved by utilizing problem-relevant representations and search operations. In the previous section, we presented an efficient representation of multidimensional block clusters. This section presents a greedy strategy to grow these clusters as key building blocks that can be incorporated into any meta-heuristic or population-based search algorithm.

The greedy stochastic search algorithm (see Algorithm 1) can be considered a hill-climbing approach to finding a local maximum cost function by alternating the identification of the categories $\mathbf{x}_j^c$ of the jth dimension based on the remaining sets $\mathbf{x}_{j*}^c = \mathbf{x}^c \setminus \mathbf{x}_j^c$ of dimensions, which can be interpreted as an expectation maximization algorithm that maximizes the conditional probability $p\left(\mathbf{x}_j^c | \mathbf{x}_{j*}^c\right)$ in every iteration.

The solution is accepted when the resulting cluster is more homogeneous than a threshold value $H^c > tr$. This greedy neighborhood search meta-heuristic method is similar to the DBSCAN (Density-based spatial clustering of applications with noise) algorithm that looks for points with many nearby neighbors (Schubert, Sander, Ester, Kriegel, & Xu, 2017). As the these algorithms proceed by arbitrarily picking up a point in the dataset (until all points have been visited) or the starting point is selected at random, most of these methods are stochastic.

Such hill-climbing-based stochastic algorithms can converge to local maxima. The problem of local minima can be handled by using diversification heuristics such as (1) the use of multiple initialization and (2) the application of population-based search heuristics, such as randomly sampling and mixing the solutions. The number of these extensions is almost infinite. Due to the limits of this paper, we do not cover all of these extensions, so in the following, the efficiency of the core greedy search with the following important extensions will be presented:

- Merging overlapping clusters.
- Removing small clusters representing noise.
- Randomly initializing new clusters.

Algorithm 1: The core steps of the proposed multidimensional multiblock clustering (MDMBC) algorithm family

```
Result: Cluster blocks of high-dimensional data
Initialization based on randomly selected samples as cluster cores;
while while the clusters do not change do
    for each cluster do
        for each dimension (in random order) do
            estimate $\mathbf{x}_j^c$ based on the other dimensions $\mathbf{x}_{j*}^c = \mathbf{x}^c \setminus \mathbf{x}_j^c$ ;
            form the potential new cluster and calculate its
              homogeneity $H^c$ ;
            if $H^c > tr$ then
                keep the new cluster by updating $\mathbf{x}_j^c$ ;
            end
        end
    end
    Merge redundant/overlapping clusters where $sim(\mathbf{x}^c, \mathbf{x}^p) > tr_s$ ;
    Delete small stable clusters (representing noise in the data) where
      $N^c < N^{min}$ for stable (not growing) clusters;
    Randomly generate new cluster seeds if needed
end
Check the statistical significance of the identified clusters
```

The algorithm has the following hyper-parameters. The most important parameter is tr that defines minimal homogeneity ("purity") of the clusters (see Eq. (7)). As a detailed case study in Section 3.1 presents, the increase of the required purity provides cleaner clusters. This requirement usually constrains the increase of the clusters, so the number of the clusters will be increased, and the resulted clusters will be smaller. In extreme cases, noisy clusters can even disappear, as clusters that are too small or have purity lower than the requirement are left out of the cluster set. As the clusters are expanded recursively, the number of the initial clusters does not significantly influence the result when similar clusters are merged. When there is no need to parallelize the algorithm, the procedure can start with one cluster, and a new cluster is formed only when the cluster cannot be further extended.

2.4. Statistical tests for cluster validation

In the biclustering process, the submatrix and the remaining rows and columns can be compared by two-sample tests, such as the t-test and F-test. These tests can be performed for all dimensions (both for rows and columns). We extended these tests to the nth-dimensional environment. Fig. 2 shows how we can specify the comparable subsets. Comparison tests (such as the t-test and F-test) are performed between the elements of the block and the block c of the ith-dimensional negations represented by the vector $\mathbf{x}_i^{\bar{c}}$ in Fig. 2). The proposed dimensional statistical tests are used to qualify the blocks found. A two sample t-(F-test) is used to test the expectation that the mean (variance) of the original values within the selected blocks is greater than the mean (variance) values within the blocks specified by dimensional negations.

Definition 1. $\mathbf{Z}^c$ denotes the found block c represented by the binary vector $\mathbf{x}^c$. $\mathbf{Z}_i^{\bar{c}}$ denotes its ith block, represented by $\mathbf{x}_i^{\bar{c}}$. The t-test statistics for dimension i are calculated as follows:

$$t_i = \frac{m^c - m_i^{\bar{c}}}{\sqrt{\frac{S_i^2}{N^c} + \frac{S_i^2}{N_i^{\bar{c}}}}}, \tag{8}$$

where m^c is the mean value of block c and $m_i^{\bar{c}}$ is the mean value of the ith dimensional negation of block c. N^c is the number of cells in block c and $N_i^{\bar{c}}$ is the number of ith dimensional negations of block c. S^2 is an estimator of the common variance of the two samples:

$$S_i^2 = \frac{\sum (z - m^c)^2 \sum (z - m_i^{\bar{c}})^2}{N^c + N_i^{\bar{c}} - 2}, \tag{9}$$

where z is a cell of $\mathbf{Z}$.

The number of degrees of freedom for the ith dimensional t-test is specified as follows:

$$df_i = N^c + N_i^{\bar{c}} - 2 \tag{10}$$

Considering the ith dimensional negations and following Eq. (8), any kind of two-sample test can be specified. Similar to the biclustering methods, in this study, two sample t- and F-tests are used for the dimensional significance test of the found block. Similar to the biclustering method, *a block is considered a significant block if all dimensional tests are significant*.

3. Validation and application examples

Since only bi- and tri-clustering algorithms exist, we compared the proposed algorithm with the most similar method, iBBiG (Gusenleitner et al., 2012). To compare our method to the iBBiG algorithm, the 2-dimensional validation example chosen was the simulation data published by Gusenleitner et al. (2012) on their paper and the iBBiG R package (Gusenleitner & Culhane, 2019). A more detailed example that demonstrates the applicability of the method is based on the four-dimensional data of Leiden 2017's higher education ranking system.

3.1. Validation and demonstration of the main characteristics of the proposed algorithm

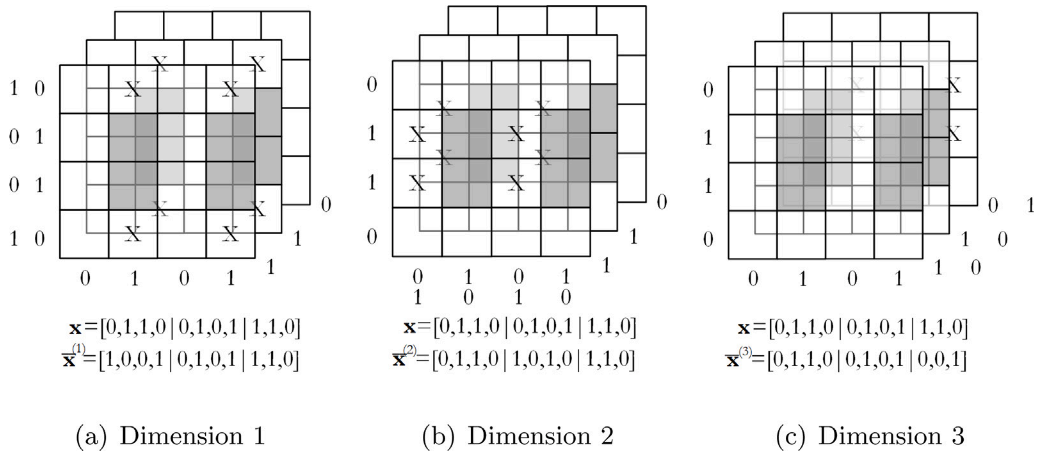
The studied benchmark dataset simulates 400 pairwise tests by 400 gene sets, in which there are seven modules (see Fig. 4(a)). Four modules overlap, and three of them are separated. In Fig. 3(a), the pure blocks (without noise) are shown. The proposed method in a two-dimensional dataset was compared by the most similar bi-clustering algorithm (Gusenleitner et al., 2012). Similar to the proposed algorithm, iBBiG also binarizes the dataset with reference to a specified threshold. The cost functions are also similar; iBBiG minimizes the entropy within a block while maximizing the size of the blocks (Gusenleitner et al., 2012). However, iBBiG cannot control the purity of a block, and the iBBiG algorithm can find only one block at each iteration, while the proposed algorithm seeks blocks simultaneously.

The benchmark problem was taken from Ref. Gusenleitner et al. (2012) (see Fig. 4(a)), where seven blocks are specified. We also ran both algorithms on cleaned and noisy data (see Fig. 3). As the purity of the biclusters significantly influences the interpretation and applicability of the results, the purity of the biclusters was investigated in addition to the size.

First, we compared the proposed algorithm with the iBBiG biclustering algorithm on two-dimensional data. As Fig. 3 shows, in the case of no noise, iBBiG and MDMBC provide similar results.

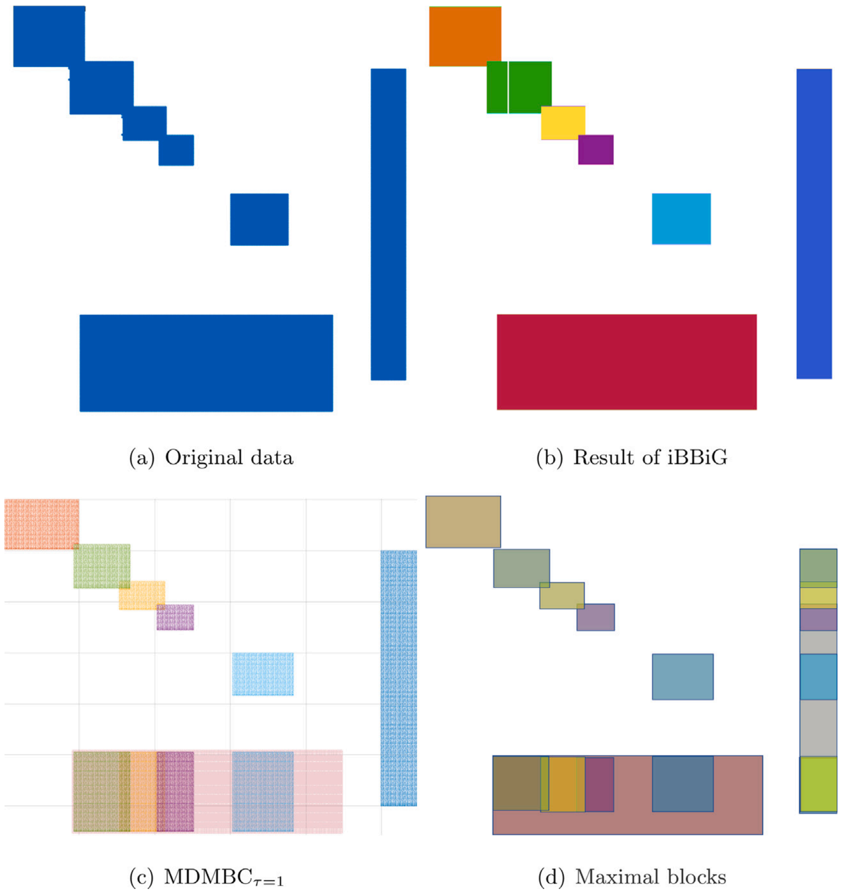
The main difference in the results is that the MDMBC selects parallel biclusters; therefore, the biclusters found are larger (see Table 1) and overlap more (see Fig. 3(c-d)). The *maximal blocks* in 3(d)) show biclusters, where the size of the clusters are maximal. The iBBiG algorithm is a subtractive algorithm, therefore, on one hand it allows overlaps, but on the other hand its subtractive nature avoids the real overlaps between clusters (see Fig. 3(b). While the proposed algorithm explores how clusters can be extended in every dimension, so that the resulting clusters are deleted may be larger than the iBBiG's clusters. These larger clusters are better if our goal is to characterize the clusters individually, as they cover all the objects that are similar in terms of the features represented by the clusters.

The greedy search can be based on any quantitative indicator, such as entropy. Most of these measures are correlated and rank the possible cluster extensions identically, so the resulted clusters will be identical. The clusters can also be compared by calculating the Jaccard distance between the binary cluster representations of different clustering results. Table 1 shows the purity and entropy of the clusters and the


$\mathbf{x}$=[0,1,1,0 | 0,1,0,1 | 1,1,0]
$\bar{\mathbf{x}}^{(1)}$=[1,0,0,1 | 0,1,0,1 | 1,1,0]
$\mathbf{x}$=[0,1,1,0 | 0,1,0,1 | 1,1,0]
$\bar{\mathbf{x}}^{(2)}$=[0,1,1,0 | 1,0,1,0 | 1,1,0]
$\mathbf{x}$=[0,1,1,0 | 0,1,0,1 | 1,1,0]
$\bar{\mathbf{x}}^{(3)}$=[0,1,1,0 | 0,1,0,1 | 0,0,1]

(a) Dimension 1 (b) Dimension 2 (c) Dimension 3

Fig. 2. Original block (see gray cells) and dimensional negations (see “X” cells).



(a) Original data (b) Result of iBBiG (c) MDMBC$_{\tau=1}$ (d) Maximal blocks

Fig. 3. Results of bi-clustering without noise.

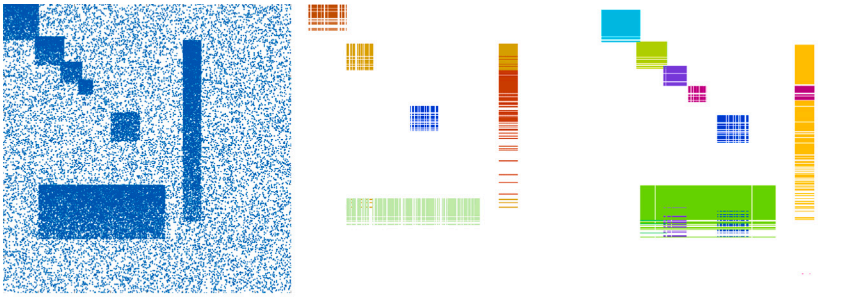
Jaccard distance from the maximal blocks. The results shows, that the MDMBC finds larger blocks than iBBiG has, which are closer to maximal blocks and the covered blocks of MDMBC are larger, cleaner and more overlapped.

The user of the proposed algorithm can specify the minimal purity of the clusters (see Eq. (7)). The increase of the required purity provides biclusters that are cleaner but smaller. The Table 2 shows, that MDMBC with $\tau > 0.75$ constraints finds cleaner and bigger blocks, while the distances from the maximal blocks are lower.

As this example demonstrates, the proposed algorithm has similar performance to the widely applied iBBiG; however, it has some benefits, as it finds clusters (blocks) that are larger in every possible direction of the selected features, and by controlling the required block purity and the minimal size of the blocks, users can fully control what kinds of blocks are identified from the data.

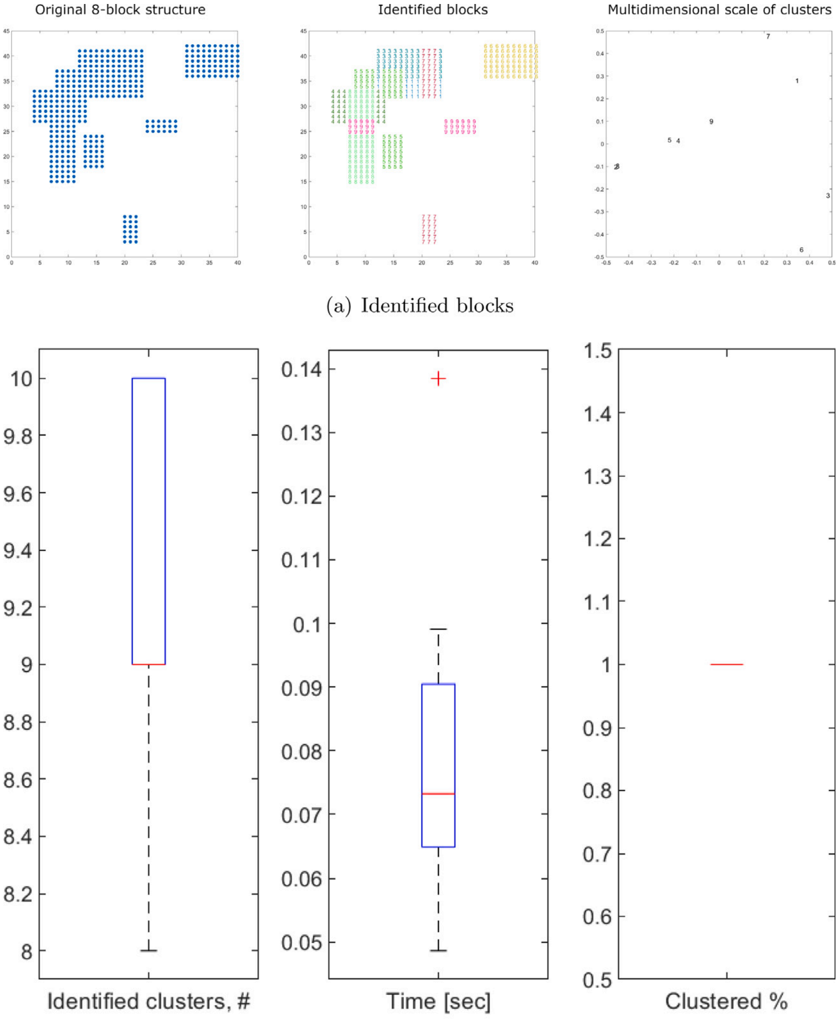
While both iBBiG and the MDMBC finds 7 blocks, blocks 5 and 7 of iBBiG differ from the original blocks (see Table 2. As Fig. 4(c) show the algorithm can identify overlapping clusters (see the rectangles with overlapping corners). This beneficial property of the algorithm results illustrate larger clusters can be covered by a set of overlapping blocks.

A multidimensional test problem generator has been developed to test how the dimensionality of the data and the selection of the initial cluster cores influence the results.



(a) Original data (b) Result of iBBiG (c) $\mathrm{MDMBC}_{\tau>0.75}$

Fig. 4. Results of bi-clustering with noise.



(a) Identified blocks

(b) Results of individual runs

Fig. 5. Box plots representing the performance of 10 independent runs on the presented 2-dimensional clustering problem. One illustrative clustering result is depicted on the middle of (a). The right subplot at (a) shows the similarity of the identified clusters.

Firstly the stochastic nature of the search algorithm is evaluated based on the result of 10 independent runs with different initial points. Fig. 5 shows a noise-free 2-dimensional example to 8 highly overlapped blocks. In most cases 9 blocks are identified (see Fig. 5(a)). It is important to note that block 2 and block 8 overlaps (see MDS in the right side of Fig. 5(a); therefore, MDMBC correctly identifies the 8 blocks. In addition, the algorithm matched all individual cells into an adequate block. The runtimes are also depicted in Fig. 5(b), which shows that the algorithm finds the clusters in less than 0.1 s.

The effect of the increase of the dimensionality of the dataset was examined by extending the dimensionality of the problem described above (see Fig. 5). In this case, 8 n-dimensional block are generated in a 50 variable/dimension space. The generated data were also analyzed in 10 independent runs. Fig. 6 shows the mean values of the indicators calculated during these runs.

Table 1
Comparison of the results of iBBiG and MDMBC on two-dimensional benchmark data.

CL	Score (Size)		Purity (τ)		Entropy		Jaccard distance	
	iBBiG	MDMBC$_{\tau=1.0}$	iBBiG	MDMBC$_{\tau=1.0}$	iBBiG	MDMBC$_{\tau=1.0}$	iBBiG	MDMBC$_{\tau=1.0}$
1	5 976	5 976	1.00	1.00	0.00	0.00	0.00	0.00
2	2 352	6 112	1.00	1.00	0.00	0.00	0.48	0.16
3	530	4 403	1.00	1.00	0.00	0.00	0.67	0.19
4	1 640	3 120	1.00	1.00	0.00	0.00	0.49	0.18
5	4 720	4 720	1.00	1.00	0.00	0.00	0.00	0.00
6	13 834	13 952	0.96	1.00	0.04	0.00	0.00	0.00
7	717	3 648	1.00	1.00	0.00	0.00	0.48	0.19

Table 2
Comparison of the results of iBBiG and MDMBC on two-dimensional benchmark data.

CL	Score (Size)		Purity (τ)		Entropy		Jaccard distance	
	iBBiG	MDMBC$_{\tau>0.75}$	iBBiG	MDMBC$_{\tau>0.75}$	iBBiG	MDMBC$_{\tau>0.75}$	iBBiG	MDMBC$_{\tau>0.75}$
1	2857	3615	0.64	0.78	0.25	0.21	0.52	0.38
2	1824	5575	0.72	0.82	0.26	0.15	0.71	0.34
3	235	1440	0.61	0.82	0.27	0.17	0.88	0.30
4	589	792	0.71	0.80	0.26	0.20	0.61	0.21
5	11	282	0.67	0.75	0.33	0.21	0.92	0.31
6	8563	9718	0.62	0.75	0.29	0.18	0.35	0.22
7	25	2697	0.66	0.83	0.26	0.11	0.95	0.28

Table 3
Results of the dimensional significance tests. (The dimensions are: time period × scientific fields × HEIs × indicators).

tr	CL	Sizes	Dimensional significance tests							
			p-values (t-test)				p-values (F-test)			
			1	2	3	4	1	2	3	4
0.50	1	$7 \times 3 \times 44 \times 4$	–	<2e−64	<2e−64	<2e−64	–	<2e−64	<2e−64	<2e−64
	2	$7 \times 5 \times 660 \times 1$	–	–	<2e−64	<2e−64	–	–	<2e−64	<2e−64
	3	$6 \times 2 \times 236 \times 3$	<2e−64	<2e−64	<2e−64	<2e−64	<2e−64	<2e−64	<2e−64	<2e−64
	4	$7 \times 4 \times 409 \times 2$	–	<2e−64	<2e−64	<2e−64	–	<2e−64	<2e−64	<2e−64
0.75	1	$6 \times 5 \times 22 \times 1$	<2e−64	–	<2e−64	<2e−64	<2e−64	–	<2e−64	<2e−64
	2	$4 \times 2 \times 82 \times 1$	<2e−64	<2e−64	<2e−64	5.21e−10	<2e−64	<2e−64	<2e−64	<2e−64
	3	$4 \times 3 \times 56 \times 1$	<2e−64	<2e−64	<2e−64	<2e−64	<2e−64	<2e−64	<2e−64	<2e−64
	4	$6 \times 2 \times 77 \times 1$	<2e−64	<2e−64	<2e−64	<2e−64	<2e−64	<2e−64	<2e−64	<2e−64

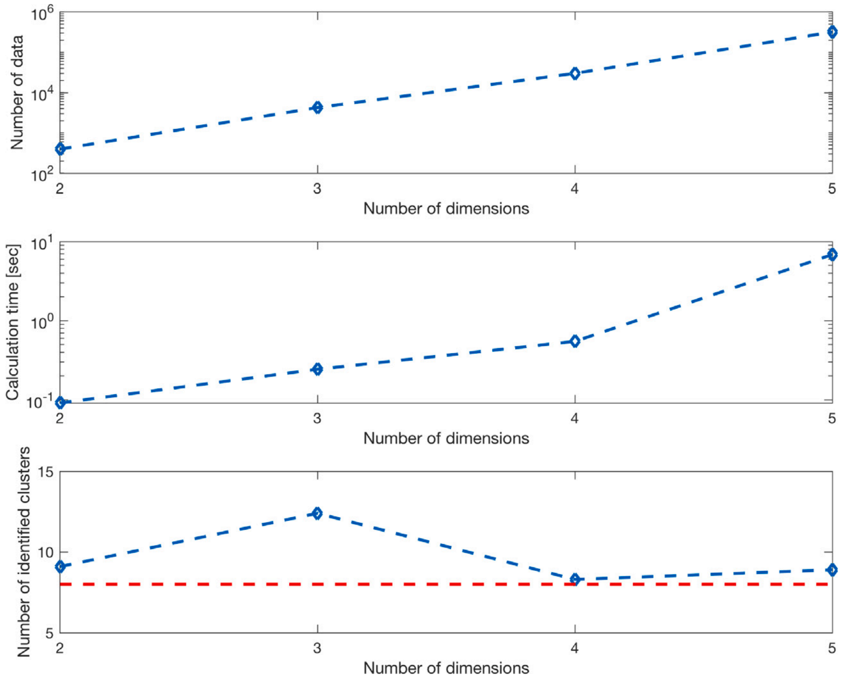
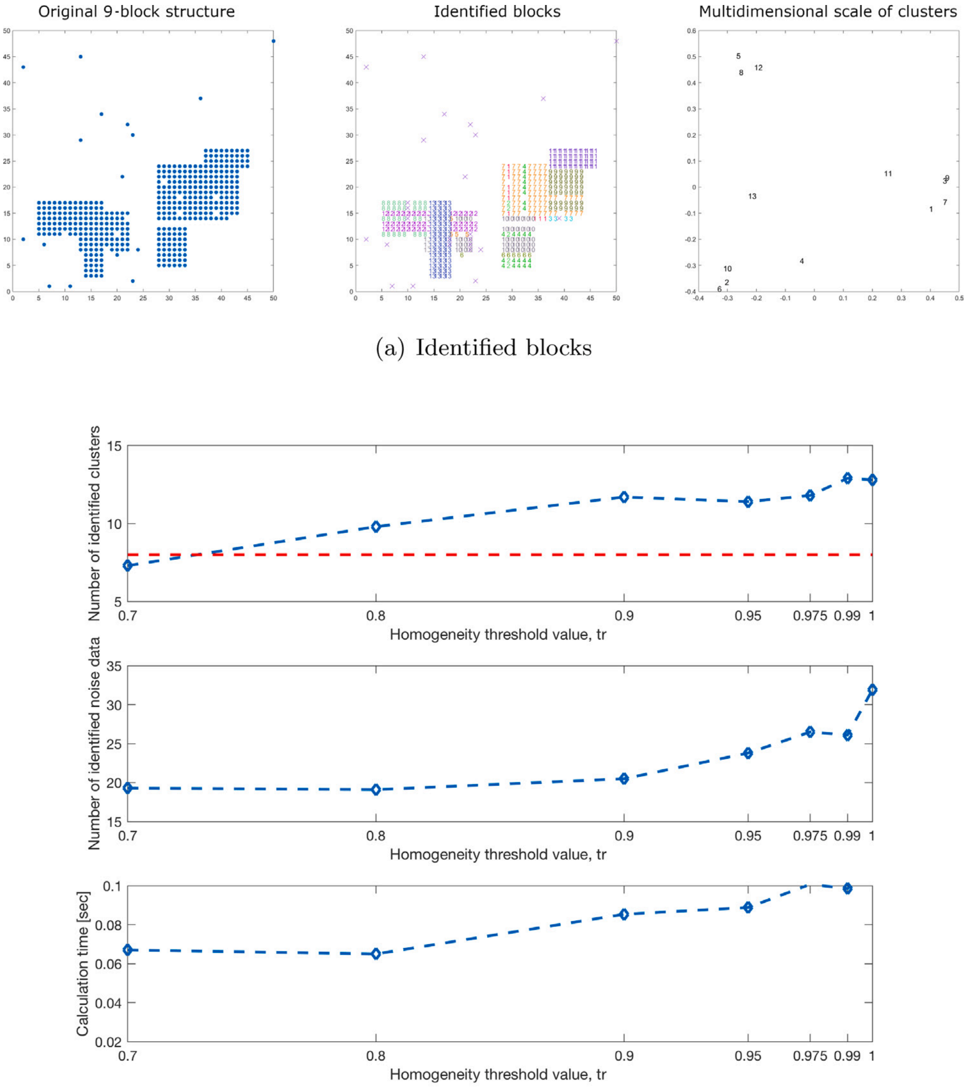


Fig. 6. The effect of the dimensionality of the dataset on the number of data, the computational time and the number of identified clusters.



(a) Identified blocks

(b) Computational and classificational results of individual runs

Fig. 7. The effect of the *tr* cluster purity threshold parameter on the number of the identified clusters and the computational time.

As the dimension increased, the number of data points increased exponentially (see the top of Fig. 6). The computational demand increases almost in proportion to the data points, which reflects that the strategy of the method is similar to the DBSCAN algorithm, which has average runtime complexity of $O(N \log N)$. In the 5-dimensional case, in 318,730 samples, the computational time did not exceed 10 s

It is important to note that the algorithm has clustered each data point for each dimension. The number of blocks only slightly exceeded 8. As Fig. 6 shows, while the number of dimensions increases, the probability of the block overlapping decreases, so the identification of blocks becomes easier.

The most important parameter of the algorithm is the cluster purity threshold value (tr). Allowing tr lower than one aims to find relevant clusters even in the case of noisy data.

For demonstration purpose, the effect of the parameter is demonstrated in a 2D example. The task is shown in Fig. 7(a) where a noise level was 1%.

The result of clustering is shown in Fig. 7(a). 10 independent tests were performed at each parameter setting, and their average is shown in Fig. 7(b).

The proposed MDMBC method, similar to the DBSCAN, identifies as noise data points that cannot be identified, which are indicated as “X” in Fig. 7(a).

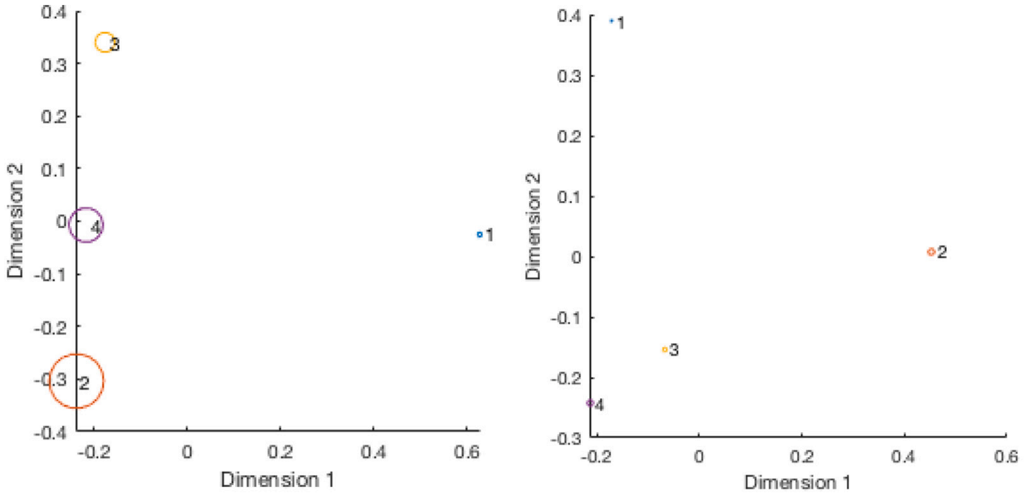
The effect of the tr parameter is shown in Fig. 7(b). Fig. 7(b) shows that group formation is more permissible at lower tr values, so fewer clusters are obtained at lower tr values than at higher tr values. As the method identifies unclassified data as noise, the number of identified blocks does not increase significantly (see Fig. 7(b)). Although the difficulty of the clustering increases with increasing tr value, this does not significantly influence the computational time.

3.2. Multiobjective analysis of higher education excellence

3.2.1. Motivation: finding multidimensional blocks of excellent universities

The proposed method is ideal for selecting institutes, scientific fields, time periods, and adequate indicators simultaneously for further exploration. An important field of application is finding institutions, indicators, and scientific fields where HEIs stand out from other HEIs.

The authors of the Leiden ranking prefer to select adequate (continuous) indicators rather than to follow one-dimensional ranking. At the



(a) $tr_{0.5}$ (Q1-Q2), $\tau_1 = 0.9424, \tau_2 = 0.9645, \tau_3 = 0.9671, \tau_4 = 0.9499$

(b) $tr_{0.75}$ (Q1), $\tau_1 = 0.9394, \tau_2 = 0.9710, \tau_3 = 0.9866, \tau_4 = 0.9784$

Fig. 8. Results of the Jaccard distance-based multidimensional scaling (MDS) of the blocks found.

Table 4
Top 10 universities in blocks ($tr = 0.5$).

Block 1	Block 2	Block 3	Block 4
Massachusetts Institute of Technology	Lille 2 University of Health and Law	London School of Hygiene & Tropical Medicine	London School of Hygiene & Tropical Medicine
Rockefeller University	China Medical University Taiwan	Rockefeller University	Harvard University
Harvard University	Taipei Medical University	University of Paris VII - Paris Diderot	University of Paris VII - Paris Diderot
University of California San Francisco	London School of Hygiene & Tropical Medicine	University of Geneva	Rockefeller University
Stanford University	National Yang Ming University	University of Amsterdam	Massachusetts Institute of Technology
California Institute of Technology	University of Paris V - Paris Descartes	King Abdullah University of Science and Technology	Weill Cornell Medical College
University of California Berkeley	University of Paris VI - Pierre and Marie Curie	Radboud University Nijmegen	Baylor College of Medicine
Princeton University	Islamic Azad University Science & Research Tehran	Leiden University	University of California San Francisco
Yale University	King Abdullah University of Science and Technology	VU University Amsterdam	Pompeu Fabra University
London School of Hygiene & Tropical Medicine	University Paul Sabatier	University of Antwerp	University of Paris VI - Pierre and Marie Curie

same time, the economic and social environments of these HEIs differ significantly from each other, and therefore, the indicator values may be very different for each HEI. Thanks to this significant difference, the binarization of the variables allows the separation of excellence and the remaining blocks. After the binarization, 1 in a cell means that a university is excellent in a given field, in a given indicator, in a given time period. Thus, the block of cells with a value of 1 identifies the group of universities that excel similarly.

The aim of the application of the proposed MDMBC algorithm is twofold. First, parallel sets of HEIs, indicators, time periods, and scientific fields have to be found that are better than the remainder. These four-dimensional blocks specify the elite “leagues of HEIs”. These blocks may be small but numerous. For specifying regional rankings, similarity is important, but if there are very few HEIs in a block, the rankings may be meaningless; therefore, the other goal is to find the smallest set of the largest leagues, where the purity, and in this way, the homogeneity, is higher than a certain threshold.

3.2.2. Details of data preprocessing and running the algorithm

The applied higher education ranking system of Leiden 2017 measures the scientific performance of 903 major higher education institutions (HEIs) using 20 indicators (see in Table 5) based on two main factors, such as the publication output and the collaboration activity in 5 scientific fields over 7 time periods. The scientific impact are calculated in two ways. In this study, only the results of the full counting method are considered, where the full counting method gives a full weight of one to each publication of a university. (see http://www.leidenranking.com/downloads).

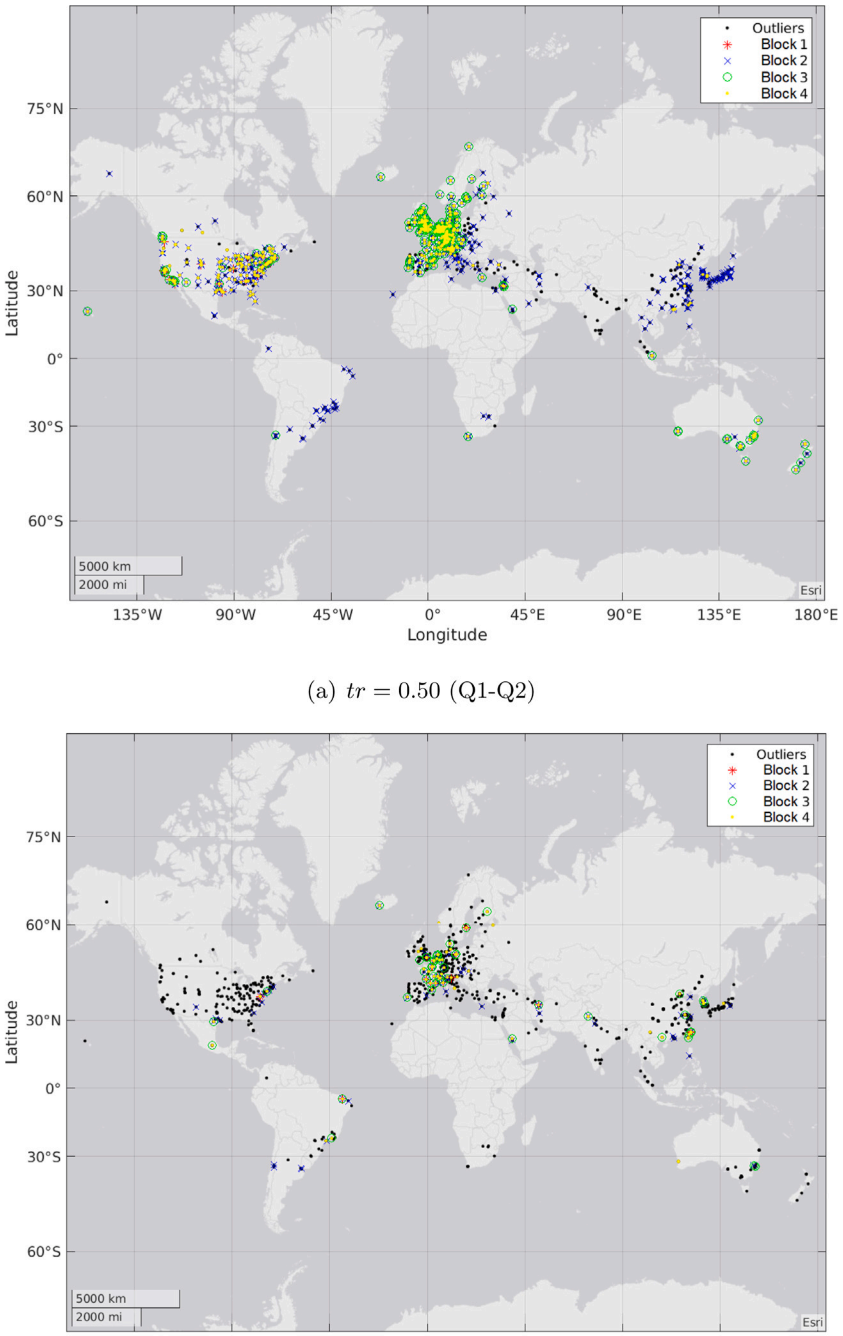
For implementation, first, every column in Leiden’s dataset is normalized with [0,1] min–max normalization to compare the different ranges of variables.

In the second step, the normalized dataset is binarized by applying two thresholds. The first threshold is set to be the median ($tr_{0.5}$, or in other words, the first two quartiles (Q1-Q2)), and the second threshold is the first quartile ($tr_{0.75}$ (Q1)). The minimal purity (τ) is 0.85, and the expected number of blocks (n_c) is 4.

3.2.3. Discussion of the results

In this run, the aim is to find the smallest number of leagues with as many HEIs as possible, where the purity and homogeneity are controlled. These blocks provide the largest blocks, showing us which HEIs can be compared with which indicators, in which profession areas, and during which times.

The Jaccard distance-based multidimensional scale (MDS) method is used to present the 4D blocks, where the diameter of the marker



(a) $tr = 0.50$ (Q1-Q2)

(b) $tr = 0.75$ (Q1)

Fig. 9. Results of the 4-dimensional multiblock clustering of the Leiden dataset.

is proportional to the size of the block (i.e. league) (see Fig. 8). The results show that the highest league contains very few HEIs. Comparing Fig. 8(a) and (b), it is seen that if the threshold is increased, the size of the blocks decreases, which means fewer leagues and fewer HEIs in leagues can be identified. The proposed algorithm identifies four blocks (see Fig. 8). Similar blocks are found if the HEIs in the first quartile are clustered ($tr_{0.75}$, see Fig. 8(b)), but these blocks contain fewer HEIs and fewer indicators (see Table 3). All dimensional tests show that the blocks found are firmly separated from the remainder. Since, in the case of ($tr = 0.5$), most of the time periods are selected, the leagues of HEIs (Q1-Q2) are stable over time.

If blocks are selected above the median ($tr_{0.5}$ (Q1–Q2)), the largest block contains most HEIs (660) in all time periods and all scientific fields; however, they are good only according to 1 indicator (pp_colab=Proportion of output resulting from scientific cooperation). In block 1, there are 44 HEIs, three scientific fields—Biomedical and health science, Life and Earth Science and Physical Sciences and Engineering—and four indicators: the proportion of a university's publications that, compared with other publications in the same field and in the same year, are in the top 10% (*PP(top 10%)*) and top 50% (*PP(top 50%)*) of the most frequently cited works; the proportion of a university's publications that were co-authored with one or more other

Table 5
Indicators of CWTS-Leiden Ranking 2017 dataset.

Scientific impact	TCS	The total	Number of citations of the publications of a university.		
	TNCS			- Normalized for field and publication year.	
	MCS	The average			
	MNCS			- Normalized for field and publication year.	
	P_top1	The number	Of a university's publications that, compared with other publications in the same field and in the same year, belong to the top	1	% most frequently cited.
	P_top10			10	
	P_top50			50	
	PP_top1	The proportion		1	
	PP_top10			10	
	PP_top50			50	
Collaboration	P_collab	The number	Of a university's publications	That have been co-authored with	With one or more other organizations.
	PP_collab	The proportion			
	P_int_collab	The number			By two or more countries.
	PP_int_collab	The proportion			
	P_industry_collab	The number			With one or more industrial organizations.
	PP_industry_collab	The proportion			
	P_short_dist_collab	The number		With a geographical collaboration distance of	Less than 100 km.
	PP_short_dist_collab	The proportion			
	P_long_dist_collab	The number			More than 5000 km.
	PP_long_dist_collab	The proportion			

organizations (*PP_collab*); and the average number of citations of the publications of a university, normalized for field and publication year (MNCS).

The block numbers are ordered by the mean value of the normalized data within the blocks. Therefore, the best universities belong to block 1. They are good according to 4 indicators in 3 scientific fields over all time intervals. Table 4 shows the first 10 institutions by the mean of Leiden's indicators. The top 10 universities of the 44 in block 1 are mainly elite US universities, such as MIT, CalTech, Harvard, Stanford, and Berkeley. While block 1 contains only 44 HEIs, one of the largest blocks (block 2) contains 660 HEIs. Although all scientific fields are included in block 2 (see Table 3, medical schools are in the top four positions. This block is more heterogeneous: in block 2, the top 10 universities are from China and European countries, such as France and England, as well as African countries.

Block 3 (236 HEIs) does not contain the first time period (2006–2009) because these are mainly emerging universities. Only two scientific fields are included, such as Life and Earth Science and Physical Sciences and Engineering. Despite this fact, the first university is the London School of Hygiene & Tropical Medicine. Three indicators belong to block 3: *PP(top 10%)*, *PP(top 50%)*, and *PP_collab*. Seven universities in block 3 come from Europe. Block 4 (409 HEIs) contains only two indicators, *PP(top 50%)* and *PP_collab*, but most scientific fields except Social Sciences and Humanities. It is interesting that the London School of Hygiene & Tropical Medicine is also the first in block 4. The top 10 HEIs in block 4 are heterogeneous; they contain mainly emerging institutions as well as several elite European, Australian, New Zealand and US universities, as shown in Fig. 9(a). The reason that many emerging universities entered this league may be that block 4 contains two fewer indicators than block 1— *PP(top 10%)* and *MNCS*—but one more scientific field: Mathematical and Computer Science. Since elite universities, such as Harvard and MIT, are involved in more than one block, they are good according to more indicators and in more scientific fields. For the time being, it is difficult for emerging universities to publish in top journals (*PP(top 10%)*), and they currently have fewer citations (*MNCS*). One emerging scientific field is Mathematical and Computer Science, because both block 2 and block 4, containing most universities, contain this scientific field.

The most important indicator is the *PP_collab* indicator, which is selected in all four blocks when the threshold is increased to 0.75 (tr = 0.75 (Q1)). Nevertheless, the four-dimensional blocks represent different fields and different time intervals in the case of Q1 leagues. Fig. 9 shows that multi-block memberships are geographically defined. Several Western European, Australian and US HEIs are separated into more than one block, while most HEIs of the BRICS (Brazilian, Russian, Indian, Chinese, and South African) countries are involved only in block 2 (tr = 0.5). They are rated as good (upper median) at pp_colab=Proportion of output resulting from scientific cooperation. Nevertheless, most of the HEIs of BRICS countries are excluded from the Q1 ($tr = 0.75$) leagues.

4. Conclusions

Multiblock clustering exhibits high potential not only in bioinformatics but also in business and social sciences fields. The proposed representation allows the identification of homogeneous and statistically significant blocks in a multidimensional dataset. The main benefit of the proposed algorithm is that it utilizes a greedy search algorithm to explore how clusters can be extended in every dimension, which makes the resulting clusters larger than those obtained by iBBiG. These larger clusters are better if we would like to characterize clusters individually, as they cover all the objects that are similar in terms of the features represented by the clusters. The proposed multidimensional multiblock clustering (MDMBC) algorithm was applied to the qualification of institutions according to the examined indicators. The proposed method identifies elite leagues of HEIs and specifies a set of indicators, set of scientific fields and set of time periods.

Although the results prove the applicability and efficiency of the method, it is known that the utilized hill-climbing-based algorithm converges to a local maximum. The problem of local minima can be handled by using diversification heuristics such as (1) the use of multiple initializations and (2) the application of population-based search heuristics, such as sampling randomly and mixing the solutions. The number of these extensions is almost infinite. Due to the limits of this paper, we did not cover these extensions, and we wanted to present how the core of the proposed algorithm family performs. The MATLAB code of the algorithm is available at the website of the authors at www.abonyilab.com, so we hope that this paper and code can serve as a starting point that motivates further research on finding clusters in multidimensional datacubes.

CRediT authorship contribution statement

Zsolt T. Kosztyán: Conception, Study design, methods used, Acquisition and collation of data, Analysis, interpretation of data, Writing the manuscript, Critical revision of paper. **András Telcs:** Critical revision of paper. **János Abonyi:** Conception, Study design, methods used, Acquisition and collation of data, Analysis, interpretation of data, Writing the manuscript, Critical revision of paper.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

Project TKP2020-NKA-10 has been implemented with the support provided by the National Research, Development and Innovation Office of Hungary, financed under the 2020-4.1.1-TKP2020 Thematic Excellence Program 2020-National Challenges sub-program funding scheme and by the Research Centre at the Faculty of Business and Economics (No. PE-GTK-GSKK A095000000-1) of the University of Pannonia (Veszprém, Hungary). The research reported in this paper was partially supported by the BME-Artificial Intelligence FIKP grant of Ministry of Human Resources (BME FIKP-MI/SC) and by the Ministry of Innovation and Technology and the National Research, Development and Innovation Office of Hungary within the Artificial Intelligence National Laboratory of Hungary.

References

Amato, R., Ciaramella, A., Deniskina, N., Mondo, C. D., di Bernardo, D., Donalek, C., et al. (2006). A multi-step approach to time series analysis and gene expression clustering. *Bioinformatics*, *22*(5), 589–596.

Ayadi, W., Elloumi, M., & Hao, J.-K. (2010). Iterated local search for biclustering of microarray data. In *IAPR international conference on pattern recognition in bioinformatics* (pp. 219–229). Springer.

Chandana, B., Srinivas, K., & Kumar, R. K. (2014). Clustering algorithm combined with hill climbing for classification of remote sensing image. *International Journal of Electrical and Computer Engineering*, *4*(6), 923.

Cheng, Y., & Church, G. M. (2000). Biclustering of expression data. In *Proceedings of the eighth international conference on intelligent systems for molecular biology* (pp. 93–103). AAAI Press.

Divina, F., & Aguilar-Ruiz, J. S. (2006). Biclustering of expression data with evolutionary computation. *IEEE Transactions on Knowledge and Data Engineering*, *18*(5), 590–602.

Dolnicar, S., Kaiser, S., Lazarevski, K., & Leisch, F. (2012). Biclustering: Overcoming data dimensionality problems in market segmentation. *Journal of Travel Research*, *51*(1), 41–49.

Forero, P. A., Baxley, P. A., & Capella, M. (2019). Co-clustering of high-order data via regularized tucker decompositions. In *ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 3442–3446). IEEE.

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

Gusenleitner, D., & Culhane, A. (2019). iBBiG: Iterative binary biclustering of genesets. R package version 1.28.0.

Gusenleitner, D., Howe, E. A., Bentink, S., Quackenbush, J., & Culhane, A. C. (2012). iBBiG: iterative binary bi-clustering of gene sets. *Bioinformatics*, *28*(19), 2484–2492.

Hatano, D., Fukunaga, T., Maehara, T., & Kawarabayashi, K.-i. (2017). Scalable algorithm for higher-order co-clustering via random sampling. In *Thirty-first AAAI conference on artificial intelligence*.

Henriques, R., & Madeira, S. C. (2018). Triclustering algorithms for three-dimensional data analysis: A comprehensive survey. *ACM Computing Survey*, (accepted paper).

Hu, J., & Pei, J. (2018). Subspace multi-clustering: a review. *Knowledge and Information Systems*, *56*(2), 257–284.

Hu, X., Zhang, H., Wu, X., Chen, J., Xiao, Y., Xue, Y., et al. (2014). A novel approach for customer segmentation based on biclustering. In Z. Huang, C. Liu, J. He, G. Huang (Eds.), *Web information systems engineering – WISE 2013 workshops* (pp. 302–312). Berlin, Heidelberg: Springer Berlin Heidelberg.

Ignatov, D. I., Gnatyshak, D. V., Kuznetsov, S. O., & Mirkin, B. G. (2015). Triadic formal concept analysis and triclustering: searching for optimal patterns. *Machine Learning*, *101*(1), 271–302.

Jain, R., Taygi, P., Sharma, M., & Khatri, S. K. (2019). Fault tolerance based indexing for multidimensional data bases. In *2019 amity international conference on artificial intelligence (AICAI)* (pp. 129–133). IEEE.

Kaiser, S., & Leisch, F. (2008). A toolbox for bicluster analysis in R.

Kaytoue, M., Codocedo, V., Buzmakov, A., Baixeries, J., Kuznetsov, S. O., & Napoli, A. (2015). Pattern structures and concept lattices for data mining and knowledge processing. In A. Bifet, M. May, B. Zadrozny, R. Gavalda, D. Pedreschi, F. Bonchi, J. Cardoso, & M. Spiliopoulou (Eds.), *Machine learning and knowledge discovery in databases* (pp. 227–231). Cham: Springer International Publishing.

Kosztyán, Z. T., Banász, Z., Csányi, V. V., & Telcs, A. (2019). Rankings or leagues or rankings on leagues? - Ranking in fair reference groups. *Tertiary Education and Management*, *25*(4), 289–310.

Lamani, A., Erraha, B., Elkyal, M., & Sair, A. (2019). Data mining techniques application for prediction in OLAP cube. *International Journal of Electrical and Computer Engineering*, *9*(3), 20–94.

Liu, J., Li, Z., Hu, X., & Chen, Y. (2009). Multi-objective ant colony optimization biclustering of microarray data. In *2009 IEEE international conference on granular computing* (pp. 424–429).

Martínez, I. N. n., Morán, J. M., & Peña, F. J. (2006). Two-step cluster procedure after principal component analysis identifies sperm subpopulations in canine ejaculates and its relation to cryoresistance. *Journal of Andrology*, *27*(4), 596–603.

Matioli, L. C., Santos, S., Kleina, M., & Leite, E. A. (2018). A new algorithm for clustering based on kernel density estimation. *Journal of Applied Statistics*, *45*(2), 347–366.

Michalak, M. (2012). Foundations of rough biclustering. In L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L. A. Zadeh, & J. M. Zurada (Eds.), *Artificial intelligence and soft computing* (pp. 144–151). Berlin, Heidelberg: Springer Berlin Heidelberg.

Michalak, M., & Ślezak, D. (2018). Boolean representation for exact biclustering. *Fundamenta Informaticae*, *161*(3), 275–297.

Michalak, M., & Stawarz, M. (2013). HRoBi–the algorithm for hierarchical rough biclustering. In *International conference on artificial intelligence and soft computing* (pp. 194–205). Springer.

Oktar, Y., & Turkan, M. (2018). A review of sparsity-based clustering methods. *Signal Processing*, *148*, 20–30.

Orzechowski, P., Sipper, M., Huang, X., & Moore, J. H. (2018). EBIC: A next-generation evolutionary-based parallel biclustering method. In *GECCO '18*, *Proceedings of the genetic and evolutionary computation conference companion* (pp. 59–60). New York, NY, USA: ACM.

Pandey, G., Atluri, G., Steinbach, M., Myers, C. L., & Kumar, V. (2009). An association analysis approach to biclustering. In *KDD '09*, *Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 677–686). New York, NY, USA: ACM.

Pontes, B., Giráldez, R., & Aguilar-Ruiz, J. S. (2015). Biclustering on expression data: A review. *Journal of Biomedical Informatics*, *57*, 163–180.

Rahaman, M. A., Turner, J. A., Gupta, C. N., Rachakonda, S., Chen, J., Liu, J. Y., et al. (2019). N-BiC: A method for multi-component and symptom biclustering of structural MRI data: Application to schizophrenia. *IEEE Transactions on Biomedical Engineering*.

Schubert, E., Sander, J., Ester, M., Kriegel, H. P., & Xu, X. (2017). Dbscan revisited, revisited: why and how you should (still) use dbscan. *ACM Transactions on Database Systems*, *42*(3), 1–21.

Strauch, M., Supper, J., Spieth, C., Wanke, D., Kilian, J., Harter, K., et al. (2007). A two-step clustering for 3-D gene expression data reveals the main features of the arabidopsis stress response. *Journal of Integrative Bioinformatics*, *4*, 81–93.

Swathypriyadharsini, P., & Premalatha, K. (2018). TrioCuckoo: A multi objective cuckoo search algorithm for triclustering microarray gene expression data. *Journal of Information Science Engineering*, *34*(6), 1617–1631.

Tanay, A., Sharan, R., & Shamir, R. (2002). Discovering statistically significant biclusters in gene expression data. *Bioinformatics*, *18*(suppl_1), S136–S144.

Veroneze, R., de França, F. O., & Zuben, F. J. V. (2011). Assessing the performance of a swarm-based biclustering technique for data imputation. In *2011 IEEE congress of evolutionary computation (CEC)* (pp. 386–393).

Vinayak, R., Hassibi, B., & EDU, C. (2016). Clustering by comparison: Stochastic block model for inference in crowdsourcing. In *Workshop machine learning and crowdsourcing*.

Wang, R., Lai, S., Wu, G., Xing, L., Wang, L., & Ishibuchi, H. (2018). Multi-clustering via evolutionary multi-objective optimization. *Information Sciences*, *450*, 128–140.

Wang, R., Miao, D., Li, G., & Zhang, H. (2007). Rough overlapping biclustering of gene expression data. In *2007 IEEE 7th international symposium on bioinformatics and bioengineering* (pp. 828–834). IEEE.