Consensus algorithm for bi-clustering analysis

Pawe Foszner[0000−0001−5491−9096], Wojciech Labaj[0000−0002−1793−9546], Andrzej Polanski[000−0001−9659−7451], and Michal Staniszewski[0000−0002−9589−2341]

Department of Computer Graphics, Vision and Digital Systems,
Faculty of Automatic Control, Electronics and Computer Science,
Silesian University of Technology,
Akademicka 2A, 44-100 Gliwice, Poland
Correspondence: pawel.foszner@polsl.pl

Abstract. Bi-clustering is an unsupervised data mining technique, which involves concurrent clustering of rows and columns of a two-dimensional data matrix. It has been demonstrated that bi-clustering may allow accurate and comprehensive mining of information, important for many practical applications. Numerous algorithms for data bi-clustering were proposed in the literature, based on different approaches and leading, in general, to different outputs. In this paper we propose a consensus method for combining outputs of many bi-clustering algorithms for improved quality of predictions. The proposed algorithm includes two steps. The first step, "assignment", leads to detecting groups of bi-clusters of high similarity and the second step, "trimming", results in transforming a group of similar bi-clusters into one bi-cluster of high quality. We demonstrate, on the basis of both simulated and real datasets, that using our algorithm highly improves quality of bi-clustering. We also provide an easy to use software tool, which includes implementations of several bi-clustering algorithms and our consensus method.

Keywords: bi-clustering · machine learning · consensus methods · ensemble methods

1 Introduction

Bi-clustering (or co-clustering) is a data analysis and data mining approach, which involves simultaneous clustering of rows and columns of a data matrix [22], [13], [21]. In recent years interest in algorithms for bi-clustering of data has substantially grown due to many new areas of applications, e.g., text analysis [9], pattern recognition [15], signal analysis [23] bioinformatics [35]. There is a large literature in the area, which can be roughly divided into parts concerning formulations of bi-clustering problems, developing bi-clustering algorithms and verifying and comparing their outcomes.

There are several versions (formulations) of bi-clustering problems. Bi-clusters can be of different types, constant values, constant rows or columns, coherent values (scaled, shifted, plaid etc.) [19]. Bi-clustering can involve continuous, discrete

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08754-7_61

or binary data. Discrete or binary bi-clusters can either originate from analyses of continuous data, where thresholding was applied as a preprocessing step [4], or can involve data of discrete origin [30]. Formulation of the bi-clustering problem also includes specifying the number of bi-clusters in the data matrix, extent of overlappings between rows and columns of bi-clusters and the level of noise in the data [1].

Several studies in the literature present surveys of bi-clustering algorithms and evaluate and compare their efficiencies [20, 26]. Methods to evaluate results of bi-clustering algorithms, in principle similar to those used when evaluating classification or clustering algorithms, can be divided into three groups, ground truth (applicable for artificially generated data, where the true structure of bi-clusters is known), internal (efficiency of the bi-clustering algorithm is measured by some quality indexes of the obtained bi-clusters, like correlations or mean square errors) and external (efficiency of the bi-clustering algorithms is measured by the significance and quality of the conclusions in the application area, e.g., biology or text processing). Compared to classification or clustering, there is an additional difficulty in evaluating quality of bi-clustering, related to more complicated output. Prelić, et al., (2006) [28] introduced a methodology for evaluating results of bi-clustering algorithms, suitable for the case where the ground truth is available, based on using a distance measure between bi-clusters given by Jaccard index and on solving the optimal assignment problem.

Each of large number of bi-clustering algorithms applied to a dataset leads, in general, to a different outcome. Estimates of bi-clusters returned by bi-clustering algorithms are sensitive to the choice of initial conditions for recursive procedures and to the choice of parameters. Therefore, an important issue becomes possibility of integrating results of different algorithms and the question whether integrating results of different algorithms can improve quality of bi-clustering. By analogy, in clustering and classification, consensus algorithms for aggregating results of clustering algorithms [34, 32] and ensemble algorithms for aggregating classification algorithms [33, 2], were proven to lead to improvements in accuracy and robustness of clustering or classification.

Surprisingly, methods for integrating/combining results of bi-clustering algorithms are very rare in the literature. We have found only one such method called ensemble method for bi-clustering, published by Hanczar and Nadif [12]. Hanczar and Nadif [12] propose the ensemble bi-clustering algorithm based on tri-clustering of auxiliary, $N \times M \times RK$ binary matrix L_{ijk} generated on the basis of outputs of R bi-clustering algorithms, each returning K bi-clusters. They demonstrate (see their figure 3) improvement of quality of bi-clustering obtained by using their ensemble method.

More detailed analysis reveals possibilities of further improvement related to some losses of information in the Hanczar and Nadif algorithm. The first information loss is related to using only binary data. Using continuous rather than binary values of the data matrix entries may potentially improve quality. The second information loss is related to neglecting the relation between bi-clusters and algorithms, which generated them. After the binary matrix L_{ijk} has

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08754-7_61

been generated there is no possibility, for any bi-cluster, to restore information on which algorithm produced it. Summing up, it seems that it is possible to construct more efficient method for integrating/combining results of bi-clustering algorithms by including information, which was neglected in the Hanczar and Nadif algorithm.

In this paper we propose a new approach for combining outputs of bi-clustering algorithms. The proposed algorithm, which we call consensus bi-clustering algorithm includes two steps. In the first step, called assignment step, the generalized assignment algorithm [27] is used to obtain groups of bi-clusters of high similarity. In the second step, called trimming step, each group of bi-clusters is turned into one bi-cluster of high quality. The first step of our algorithm utilizes information on the relations between bi-clusters and algorithms. The second step uses continuous rather than binary data, which leads to improved predictions. We demonstrate, on the basis of both simulated and real datasets, that application of our algorithm improves quality of bi-clustering compared to the method of Hanczar and Nadif. The obtained improvement increases with the increase of complexity of bi-clustering problem and with the increase of the level of noise in the bi-clustering problem.

The effectiveness of the proposed method we evaluated on specially prepared for this purpose synthetic data and adequately well-described real data from literature. Synthetic data were prepared in such a way as to cover the widest spectrum of possible scenarios occurring in bi-clustering. These cases relate to various aspects of the data matrix such as noise, number of bi-clusters, internal data structure and the degree of overlapping of clusters within both dimensions. The real data has been chosen so as to be able to clearly interpret the quality of the results obtained. Data from the work of Monica Chagoyen [5] were selected, where bi-clusters should represent groups of genes (associated with the specific gene ontology) and words. The quality of founded bi-clusters were assessed by measuring quality of assigment of bi-clusters with ontologies established at the data matrix design level.

2 Description of the proposed consensus algorithm

We assume that the desired number of bi-clusters is given, denoted by K, and that we already have outputs of L bi-clustering algorithms, each containing K (hypothetical) bi-clusters. Each of the L bi-clustering algorithms is called a component bi-clustering algorithm and bi-clusters returned by component bi-clustering algorithms are called component bi-clusters. The bi-clusters returned by our consensus algorithm are called resultant or consensus bi-clusters.

2.1 Notation

We use similar notation to that introduced by Madeira and Oliveira [19]. We denote a $n \times m$ data matrix by A. Following Madeira and Oliveira [19] we write $A = (X, Y)$, where $X = x_1, x_2, \cdots, x_n$ denotes the set of indexes of rows of A

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08754-7_61

and $Y = y_1, y_2, \cdots, y_n$ stands for the set of indexes of columns of A. One assumes that there is an underlying function $a(i, j)$, which returns entry of the matrix given indexes of its row and column, $a_{ij} = a(i, j)$. Then the precise notation would be $A = a(X \times Y)$, where $\times$ stands for Cartesian product. However, with slight abuse of notation, we shall write $A = (X, Y)$ instead of $A = a(X \times Y)$. Such notation, introduced in [19], does not lead to ambiguities and is convenient for describing bi-clusters and bi-clustering algorithms.

Bi-cluster B is defined as a pair

$$B = B_{I,J} = (I, J), I \subseteq X, J \subseteq Y. \tag{1}$$

Subsets I and J, in the above, are called sets of attributes, or attributes of the bi-cluster $B_{I,J}$. If necessary we add superscripts to index different bi-clusters, e.g., $B^{l,k} = B^{l,k}_{I,J} = (I^{l,k}, J^{l,k})$ stands for $k - th$ bi-cluster, returned by $l - th$ component bi-clustering algorithm.

The output of the l-th component bi-clustering algorithm is denoted as

$$R^l = \{B^{l,1}, B^{l,2}, \cdots, B^{l,K}\} \tag{2}$$

where

$$B^{l,i} = (I^{l,i}, J^{l,i}), \tag{3}$$

$i \in 1, \cdots, K$.

We combine outputs $R^1, R^2, ..., R^L$ of the component algorithms for data bi-clustering in two steps. In the first step, called assignment step, we use the generalized assignment algorithm [27] to obtain K groups, each containing L bi-clusters, of highest possible intra-group similarities. Jaccard index between pairs of bi-clusters is used to define the intra-group similarity measure. In the second step, called trimming step, each group of component bi-clusters obtained in the first step, is transformed into one resultant bi-cluster. Applying the second step to each of the K groups of bi-clusters results in obtaining K consensus bi-clusters.

2.2 Assignment Step

Each of the component bi-clustering algorithms returns K component bi-clusters, but no relations between bi-clusters returned by different component algorithms are known a priori. However, since component bi-clustering algorithms are applied to the same data set it is likely that the component bi-clusters will exhibit similarities stemming from the true structure of the dataset. We therefore search for K groups of component bi-clusters with possibly high intra-group similarity in each group. This problem can be formulated as a generalized assignment problem involving minimization of the index

$$I = \sum cost(G_k), \tag{4}$$

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08754-7_61

where $cost(G_k)$ is the cost of grouping component bi-clusters $B^{1,k^{(1)}}, B^{2,k^{(2)}}, \cdots, B^{L,k^{(L)}}$ into one group

$$G_k = \{B^{1,k^{(1)}}, B^{2,k^{(2)}}, \cdots, B^{L,k^{(L)}}\}, \tag{5}$$

such that no pair of the bi-clusters within the group G_k comes from the same component bi-clustering algorithm. In (5) $B^{l,k^{(l)}}$ denotes $k^{(l)}$ bi-cluster returned by the l-th component bi-clustering algorithm.

The cost function $cost(G_k)$ is computed on the basis of the Jaccard indexes between pairs of component bi-clusters. More specifically, for each pair of bi-clustering algorithms (l_1, l_2) we define a similarity matrix $S(l_1, l_2)$ between their returned sets of bi-clusters

$$S(l_1,l_2) = \begin{bmatrix} S_{Jacc}(B^{l_1,1}, B^{l_2,1}) & S_{Jacc}(B^{l_1,1}, B^{l_2,2}) & \cdots & S_{Jacc}(B^{l_1,1}, B^{l_2,K}) \\ S_{Jacc}(B^{l_1,2}, B^{l_2,1}) & S_{Jacc}(B^{l_1,2}, B^{l_2,2}) & \cdots & S_{Jacc}(B^{l_1,2}, B^{l_2,K}) \\ \cdots & \cdots & \cdots & \cdots \\ S_{Jacc}(B^{l_1,K}, B^{l_2,1}) & S_{Jacc}(B^{l_1,K}, B^{l_2,2}) & \cdots & S_{Jacc}(B^{l_1,K}, B^{l_2,K}). \end{bmatrix} \tag{6}$$

In the above $S_{Jacc}(B^{l_1,q}, B^{l_2,r})$ - denotes the value of the Jaccard similarity index between the two bi-clusters - $q'th$ bi-cluster from result l_1 and $r'th$ bi-cluster from result l_2 $(q, r = 1, \cdots, K)$. Entries of similarity matrices (6) are then used to compute the cost of grouping component bi-clusters $B^{1,k^{(1)}}, B^{2,k^{(2)}}$ $\cdots$,$\mathrm{B}^{L,k^{(L)}}$, $cost(G_k)$

$$cost(G_k) = \sum_{l_1=1}^{L} \sum_{l_2=l_1+1}^{L} S_{Jacc}(B^{l_1,k^{l_1}}, B^{l_2,k^{l_2}}). \tag{7}$$

The summation goes over all possible pairs (l_1, l_2) of component bi-clustering algorithms.

Algorithms for solving the generalized assignment problem have been developed in several papers in the literature, e.g., [27, 24]. In [27] a branch and bound algorithm for minimizing 5 was proposed. However, computational complexity of these algorithm scale exponentially with KL and therefore their application is difficult for larger datasets. There are several time efficient algorithms, which provide sub-optimal solutions to generalized assignment problems, [7, 11]. In our implementations of the consensus bi-clustering algorithm we have options of either using the branch and bound algorithm for minimizing 5 or replacing it by a suboptimal heuristic, greedy search, described in detail in the Supplementary Materials, which is much faster. In computational experiments we have verified that replacing branch and bound algorithm for minimizing formula 5 by a heuristic, greedy search does not decrease performance of the whole consensus bi-clustering algorithm.

2.3 Trimming Step

In the trimming step each group of the component bi-clusters (5) is transformed into one resultant bi-cluster B^k_{trimed}. Trimming is designed in such a way that

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08754-7_61

unreliable outputs of component bi-clustering algorithms, e.g., resulting from poorly chosen initial conditions, are corrected. One can notice that, for each group of component bi-clusters, the problem of trimming can be understood as searching for one bi-cluster in the subset of the data matrix A given by the union of component bi-clusters.

There are many possible approaches to the problem of searching for one bi-cluster in the data, e.g., [29]. In our implementation of the trimming step we use a heuristic algorithm, similar to that described in [29], designed for maximal robustness against noise in the data. The iterative procedure starts from the initial condition given by the union of all bi-clusters within G_k and is designed in such a way that the value of the ACV index (Average Corelation Value [31]) of the resulting bi-cluster must increase in each step. If there is no possibility to increase the values of thee ACV index the procedure stops.

3 Comparisons of performances of bi-clustering algorithms

We compare performance of our consensus bi-clustering algorithms to performances of component bi-clustering algorithms and to performance of Hanczar and Nadif's ensemble bi-clustering algorithm [12] for both synthetically created data and for several real datasets. For evaluation of performance of different algorithms we use ground truth method (for synthetic data) internal quality index ACV (for synthetic data and for real data) and suitably defined external indexes (for real data).

In the case of analysis of synthetic data we use a ground truth methodology described by Prelić, et al., (2006) [28] based on on solving the optimal assignment problem between the computed and the known bi-clusters in the first step and computing difference measure between bi-clusters given by the Jaccard index in the second step.

In the case of analysis of real data we use an internal quality index for evaluating quality of bi-clustering. There are several indexes suitable for evaluating quality of bi-clustering proposed in the literature e.g., mean square residue (MSR [6]), average Spearman's rho (ASR [3]), average correlation value (ACV [31]). Here we use the ACV index of the set of bi-clusters, recommended in many papers, defined as follows:

$$ACV(B) = max\left\{\frac{\sum_{i=1}^{n}\sum_{j=1}^{n}|r\_row_{ij}| - n}{n^2 - n}, \frac{\sum_{k=1}^{m}\sum_{l=1}^{m}|r\_col_{kl}| - m}{m^2 - m}\right\} \quad (8)$$

where

$$ACV(B) \in [0, 1] \quad (9)$$

Where:

– r_row_{ij} - is the value of the Pearson correlation between the $i'th$ and $j'th$ rows

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08754-7_61

– r_col_{kl} – is the value of the Pearson correlation between $k'th$ and $l'th$ columns.

The higher the value of ACV(B) the better quality of the bi-cluster B. One should also notice flexibility of the ACV index. ACV index is suitable for constant, additive and multiplicative bi-clusters.

3.1 Synthetic Data

Our aim when creating synthetic data for comparing bi-clustering algorithms was covering diversity of possible bi-clustering data. Our data sets contain every important combinations of bi-cluster structures, the degree to which overlap the rows and columns and noise level that was introduced into the bi-cluster. Data consist of matrices with one of four major structure each. Additionally every matrix represents single structure appears in one of eight variants regarding to bi-clusters overlapping over rows and columns. Finally, the last parameter used to create the data matrix was the noise level. To produce the above described matrices, BiBench software by Kemal Eren [10] was used.

To summarize and point out the details of each category we have:

1. Regarding to level of overlapping test set consist of various number for matrices with different variants of bi-clusters positions in data matrix. We distinguish data matrices with:

(a) Bi-clusters with separate sets of rows and columns,
(b) Bi-clusters with separate sets of rows and overlapping columns with different degree of overlaps (25%, 50%, 75%),
(c) Bi-clusters with separate sets of columns and overlapping rows with different degree of overlaps (25%, 50%, 75%)

2. Each structure described above appears in four different variants of bi-clusters values. Regarding to bi-clusters structure we distinguish data with (every single matrix contains only one of the following):

– Constant data
– Constant data up-regulated
– Plaid data
– Shift and scale data

3. The last, third dimension of parameters is the noise level that has been introduced into the bi-cluster. This value comes from the set (0, 0.001, 0.25, 0.5, 1)

4. Number of bi-clusters in the range $< 2 - 10 >$

All matrices are of size 500x500. The background noise is generated as a uniformly distributed random variable in the range $< 0 - 100 >$.

To sum this up we have sixteen different data sets regarding to bi-cluster position and four regarding to bi-cluster structure. Also, each matrix is present in five versions depending on the level of noise introduced into the bi-clustering. Noise is introduced to bi-cluster accordingly to BiBench algorithm. Finally, all

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08754-7_61

these variants come in 9 different versions due to the number of bi-clusters. The final set consisting of 2880 matrices, each having a different structure, different distribution and number of the bi-clusters inside data matrix and different noise level.

3.2 Text Mining Data

We have analyzed the text mining dataset, which we have created using the same scenario as described in the article by Monica Chagoyen, et al [5]. Authors of [5] analyzed 7080 scientific papers from the PubMed database devoted to genetics of the Saccharomyces cerevisiae species. To each of the papers they assigned a set of genes, selected from the lists of 575 genes from Saccharomyces cerevisiae genome, using a semi-automatic procedure supported by human experts.

Functions of genes analyzed in [5] were also summarized by eight broad biological processes described by the following GO Ontology terms: cell cycle (GO:0007049), cell wall organization and biogenesis (GO:0007047), DNA metabolism (GO:0006259), lipid metabolism (GO:0006629), protein biosynthesis (GO:0042158), response to stress (GO:0006950), signal transduction (GO:0007165), transport (GO:0006810).

We implemented the algorithm described in [5] including the following steps: 1. Downloading 7080 texts of scientific papers using lists of their PubMed ids published by [5]; 2. Extracting words in these papers; 3. Applying filters removing colloquial (most frequent, frequency $> 80\%$) and vary rare words (frequency $< 4\%$). As a result we got a word - occurrences matrix, whose rows corresponded to genes and columns to word. Entries of this matrix are balanced term frequencies, D_{ij}, of term j in document i, defined as:

$$D_{ij} = tf_{ij} * IDF_j \tag{10}$$

In the above formula IDF_j stands for the inverse document frequency of term j [48],

$$IDF_j = log\left(\frac{T}{t_j}\right) \tag{11}$$

where:

- T – total number of documents in set,
- t_j – number of documents that contains document j

When analyzing the word - occurrences matrix we search for 8 bi-clusters, each corresponding to on of 8 GO terms of eight broad biological processes, listed above.

4 Results and discussion

For a single data matrix all single results are given to the input of the consensus algorithm described in section 2, as well as the input of the tri-clustering algorithm [12]. Synthetic data were assessed both from the perspective of the quality

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08754-7_61

of the resulting bi-clusters and their mapping in a set of expected bi-clusters. Real data were evaluated using only quality index [31]. The similarity measure is obtained as follows:

- Expected and Founded set are presented as cost matrix where each row is represented by different bi-cluster form expected set and each column by bi-cluster from founded set. The values of this matrix are Jaccard indices computed on both clusters:

$$c_{i,j} = \frac{\frac{I^i \cap I^j}{I^i \cup I^j} + \frac{J^i \cap J^j}{J^i \cup J^j}}{2} \tag{12}$$

- Hungarian algorithm is performed on this cost matrix in order to find perfect assignment
- Sum of Jaccard indexes pointed out in previews point is divided by the number of founded bi-clusters

Both similarity measure and quality index range from $< 0 - 1 >$

4.1 Synthetic data

The advantage of having synthetic data is the fact that we have ground truth. This allows for calculating accurate quality measures based on it. To compare the results of synthetic data, we decided to use independent measures of quality usually used for synthetic data: Recovery and Relevance. Measures were introduced by Kemal Eren [10] and are quite useful in terms of comparing data that provides ground truth.

$$Re(R^1, R^2) = \frac{1}{|R^1|} \sum_{b_1 \in R^1} \underset{b_2 \in R^2}{max} S_{Jacc}(b_1, b_2) \tag{13}$$

Eq.13 apply for both Recovery and Relevance. It takes founded set of bi-clusters and expected set of bi-clusters at the input. Recovery: Re(Expected, Founded) can be interpreted as checking if the algorithm found all of the expected bi-clusters. Relevance: Re(Founded, Expected) can be interpreted as checking if all the found bi-clusters were expected. Both measures take values from the range $< 0, 1 >$. Figure 1 presents a graphical summary of the entire synthetic data set. A single point on the chart means the average value for Recovery and Relevance from the point of view of a single algorithm for all synthetic matrices.

For comparison have been selected algorithms specializing in different types of input data, as well as algorithms with different assumptions as to data representation. The final set of methods was as follows: BiMax [28], Floc [36], ITL [8], QUBIC [18], Triclustering [12], Consensus, Plaid [17], Cheng and Church [6], kSpectral [16], xMOTIFs [25]. The above list presents a wide spectrum of approaches in the field of bi-clustering. The results for classic methods were obtained using the implementation from the matlab mtba package [14]. The results for the ensemble methods (Triclustering and Consensus) were obtained by providing the results of the other methods on their input. It is clear here that the

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08754-7_61

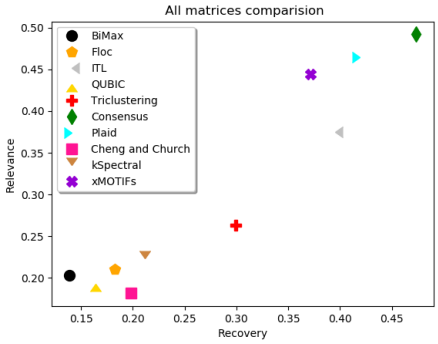


Fig. 1: Averaged results from the entire synthetic data set.

method proposed in this work improves significantly the outcome based on the results of specialized methods.

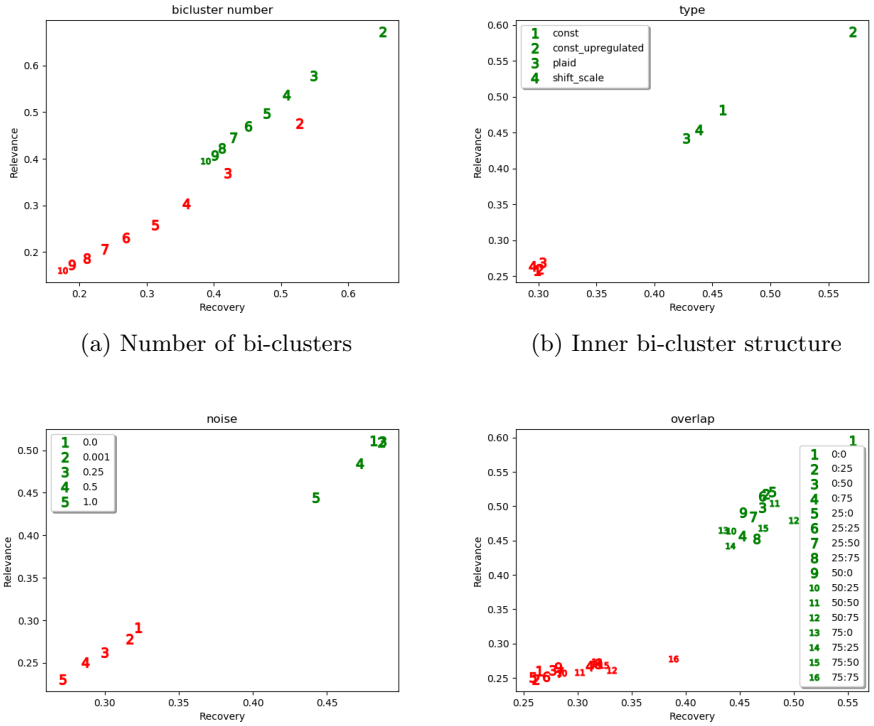
In the Figure 1 we can distinguish the following groups of algorithms: (1) in the lower left corner we have algorithms performed for a specific data type (BiMax, Floc, QUBIC, Cheng&Church, kSpectral). Therefore, on average for all cases, they do not work well. Then just above them we see firts ensemble method - Triclustering, which can improve on average for most of the results. Next we see the methods (xMOTIFs, ITL, Plaid) that do quite well regardless of the given data structure, noise, etc. On top of this we have our own Consensus algorithm that improves the outcome regardless of the quality of the results given at its input.

Figure 2 shows a detailed comparison between the algorithm proposed by Hanchar [12] (red) and the algorithm proposed in this paper (green). In the this figure we can read a few things that were to be expected. Such as increasing noise or increasing bi-clusters numbers, which reduces quality. The expected effect was also a very high result for "const_upreguleted" data. The bi-clusters there are easily found because they can be seen even on heat maps of the data matrix. It clearly shows that regardless of the aspect selected, the algorithm proposed in this work achieves better results. All results with input data matrices have been published on the dedicated website https://aspectanalyzer.foszner.pl/

4.2 Real data

Obtained bi-clusters of genes for each bi-clustering method were compared biologically based on the Gene Ontology database [1]. Along with the data there were 8 GO terms from the biological process ontology provided, which served us as a reference for 8 bi-clusters (one GO term per one bi-cluster). In order to compare our results with the reference GO terms, first a functional analysis was performed for each genes bi-cluster using the elimination algorithm [2] and Fishers exact test. The elimination algorithm was used to reduce the number of

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08754-7_61



(a) Number of bi-clusters

(b) Inner bi-cluster structure

(c) Bi-cluster noise

(d) Overlap level in percentage rows:columns

Fig. 2: Detailed comparison of Consensus and Triclustering due to various aspects of the data matrix

redundant GO terms, by taking into account only GO terms with the largest information content (IC). The level of statistical significance was set at 0.05. Received list of statistically significant GO terms for each bi-cluster was compared with each of the reference GO term. For this purpose a semantic similarity measure, Wang method [3] which takes into account hierarchy of GO terms in the ontology tree, was used. Next, combining of semantic similarity measures was performed, checking for each reference GO term and each bi-cluster of GO terms what is the maximum value of similarity measure. As a result, a similarity matrix was obtained where our result bi-clusters were in the rows and reference bi-clusters in the columns. Finally, the Hungarian algorithm was used to assign bi-clusters to each other, maximizing the measure of similarity. By summing up the received measures for all bi-clusters, we obtain a measure of quality, which takes into account the biological information flowing from the data.

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08754-7_61

Table 1: A result table showing how the ontological terms were recreated by our algorithm and the Hanczar algorithm. Value describe in section 4.2

	Consensus	**Triclustering**
GO:0007049	0.872	0.698
GO:0071555	1	0
GO:0006259	1	0.944
GO:0006629	1	0
GO:0042158	0.648	0.45
GO:0006950	0.584	0.643
GO:0007165	0.875	1
GO:0006810	0.632	0
SUM	6.611	3.735

5 Conclusions

Hanczar and Nadif [12] integrate outputs of component bi-clustering algorithms by using the procedure of binary tri-clustering, which may lead to loss of some information. In our algrithm we assign the weight to each attribute of the group of component bi-clusters, which reflects its potential influence on the quality of the resulting bi-cluster. In the iterative procedure of consensus trimming we remove attributes characterized by least weights.

As shown by numerical results of Section 4 of the consensus algorithm is much better than direct competition in the form of Tri-clustering algorithm. First of all, it was proved that the proposed method meets the basic assumptions for ensemble methods. As Figure 1 shows, the result was better than any of the methods given on its input. Next results were checked in detail for various aspects of the data matrix. For noise, the number of bi-clusters, internal structure of bi-cluster and for the cluster overlapping rate - the proposed method achieved very good results (Figure 2). Finally, the method was evaluated for real data. We selected well described in the literature text mining data, where bi-clusters consist of words and genes. The quality of the method was assessed based on a comparison of the results with the original intentions of the creators of the data matrix. As Table 1 shows, the results were twice as good as the other ensemble method.

As for the development of the method described here, it has two aspects. The first is to make this method available as a service. It will be possible to perform a bi-clustering experiment with algorithms from the literature, as well as conduct a consensus (ours or Hanczar). On the scientific level, we would like to focus on the automatic recognition of the number of bi-clusters present in the input data. We believe that it can be done by analyzing the quality of bi-clusters from various executions.

All and detailed results can be found at https://aspectanalyzer.foszner.pl

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08754-7_61

6 Acknowledgements

This work was funded by Polish National Science Centre, OPUS grant 2016/21/B /ST6/02153 (AP,WL); research project (RAU-6, 2020) and projects for young scientists of the Silesian University of Technology (Gliwice, Poland) (PF,MS)

References

1. Aguilar-Ruiz, J.S.: Shifting and scaling patterns from gene expression data. Bioinformatics **21**(20), 3840–3845 (2005)
2. Avidan, S.: Ensemble tracking. Pattern Analysis and Machine Intelligence, IEEE Transactions on **29**(2), 261–271 (2007)
3. Ayadi, W., Elloumi, M., Hao, J.K.: A biclustering algorithm based on a bicluster enumeration tree: application to dna microarray data. BioData mining **2**(1), 9 (2009)
4. Benabdeslem, K., Allab, K.: Bi-clustering continuous data with self-organizing map. Neural Computing and Applications **22**(7-8), 1551–1562 (2013)
5. Chagoyen, M., Carmona-Saez, P., Shatkay, H., Carazo, J.M., Pascual-Montano, A.: Discovering semantic features in the literature: a foundation for building functional associations. BMC bioinformatics **7**(1), 1 (2006)
6. Cheng, Y., Church, G.M.: Biclustering of expression data. In: Ismb. vol. 8, pp. 93–103 (2000)
7. Cohen, R., Katzir, L., Raz, D.: An efficient approximation for the generalized assignment problem. Information Processing Letters **100**(4), 162–166 (2006)
8. Dhillon, I.S., Mallela, S., Modha, D.S.: Information-theoretic co-clustering. In: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 89–98. ACM (2003)
9. Diaz, A.K.R., Peres, S.M.: Biclustering and coclustering: concepts, algorithms and viability for text mining. Revista de Informática Teórica e Aplicada **26**(2), 81–117 (2019)
10. Eren, K., Deveci, M., Küçüktunç, O., Çatalyürek, Ü.V.: A comparative analysis of biclustering algorithms for gene expression data. Briefings in bioinformatics **14**(3), 279–292 (2013)
11. Fleischer, L., Goemans, M.X., Mirrokni, V.S., Sviridenko, M.: Tight approximation algorithms for maximum general assignment problems. In: Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm. pp. 611–620. Society for Industrial and Applied Mathematics (2006)
12. Hanczar, B., Nadif, M.: Ensemble methods for biclustering tasks. Pattern Recognition **45**(11), 3938–3949 (2012)
13. Hartigan, J.A.: Direct clustering of a data matrix. Journal of the american statistical association **67**(337), 123–129 (1972)
14. J. K. Gupta, S. Singh, N.K.V.: Mtba: Matlab toolbox for biclustering analysis. pp. 94–97. IEEE (2013)
15. Kerr, G., Ruskin, H.J., Crane, M., Doolan, P.: Techniques for clustering gene expression data. Computers in biology and medicine **38**(3), 283–293 (2008)
16. Kluger, Y., Basri, R., Chang, J.T., Gerstein, M.: Spectral biclustering of microarray data: coclustering genes and conditions. Genome research **13**(4), 703–716 (2003)
17. Lazzeroni, L., Owen, A.: Plaid models for gene expression data. Statistica sinica pp. 61–86 (2002)

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08754-7_61

18. Li, G., Ma, Q., Tang, H., Paterson, A.H., Xu, Y.: Qubic: a qualitative biclustering algorithm for analyses of gene expression data. Nucleic acids research p. gkp491 (2009)
19. Madeira, S.C., Oliveira, A.L.: Biclustering algorithms for biological data analysis: a survey. IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB) **1**(1), 24–45 (2004)
20. Maind, A., Raut, S.: Comparative analysis and evaluation of biclustering algorithms for microarray data. In: Networking communication and data knowledge engineering, pp. 159–171. Springer (2018)
21. Mirkin, B.: Mathematical classification and clustering, volume 11 of nonconvex optimization and its applications (1996)
22. Morgan, J.N., Sonquist, J.A.: Problems in the analysis of survey data, and a proposal. Journal of the American statistical association **58**(302), 415–434 (1963)
23. Moussaoui, S., Hauksdottir, H., Schmidt, F., Jutten, C., Chanussot, J., Brie, D., Douté, S., Benediktsson, J.A.: On the decomposition of mars hyperspectral data by ica and bayesian positive source separation. Neurocomputing **71**(10), 2194–2208 (2008)
24. Munkres, J.: Algorithms for the assignment and transportation problems. Journal of the Society for Industrial and Applied Mathematics **5**(1), 32–38 (1957)
25. Murali, T., Kasif, S.: Extracting conserved gene expression motifs from gene expression data. In: Biocomputing 2003, pp. 77–88. World Scientific (2002)
26. Padilha, V.A., Campello, R.J.: A systematic comparative evaluation of biclustering techniques. BMC bioinformatics **18**(1), 55 (2017)
27. Pierskalla, W.P.: Letter to the editorthe multidimensional assignment problem. Operations Research **16**(2), 422–431 (1968)
28. Prelić, A., Bleuler, S., Zimmermann, P., Wille, A., Bühlmann, P., Gruissem, W., Hennig, L., Thiele, L., Zitzler, E.: A systematic comparison and evaluation of biclustering methods for gene expression data. Bioinformatics **22**(9), 1122–1129 (2006)
29. Rangan, A.V.: A simple filter for detecting low-rank submatrices. Journal of Computational Physics **231**(7), 2682–2690 (2012)
30. Rodriguez-Baena, D.S., Perez-Pulido, A.J., Aguilar, J.S., et al.: A biclustering algorithm for extracting bit-patterns from binary datasets. Bioinformatics **27**(19), 2738–2745 (2011)
31. Teng, L., Chan, L.: Discovering biclusters by iteratively sorting with weighted correlation coefficient in gene expression data. J. Signal Process. Syst. p. 15201527 (2010)
32. Topchy, A., Minaei-Bidgoli, B., Jain, A.K., Punch, W.F.: Adaptive clustering ensembles. In: Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on. vol. 1, pp. 272–275. IEEE (2004)
33. Tsoumakas, G., Vlahavas, I.: Random k-labelsets: An ensemble method for multilabel classification. In: Machine learning: ECML 2007, pp. 406–417. Springer (2007)
34. Vega-Pons, S., Ruiz-Shulcloper, J.: A survey of clustering ensemble algorithms. International Journal of Pattern Recognition and Artificial Intelligence **25**(03), 337–372 (2011)
35. Xie, J., Ma, A., Zhang, Y., Liu, B., Cao, S., Wang, C., Xu, J., Zhang, C., Ma, Q.: Qubic2: a novel and robust biclustering algorithm for analyses and interpretation of large-scale rna-seq data. Bioinformatics **36**(4), 1143–1149 (2020)
36. Yang, J., Wang, H., Wang, W., Yu, P.S.: An improved biclustering method for analyzing gene expression profiles. International Journal on Artificial Intelligence Tools **14**(05), 771–789 (2005)

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08754-7_61