Pattern Recognition 104 (2020) 107318

Contents lists available at ScienceDirect

Pattern Recognition

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

Biclustering with dominant sets

M. Denitto[a,*], M. Bicego[a], A. Farinelli[a], S. Vascon[b], M. Pelillo[b]

[a] University of Verona, Verona, Italy
[b] ECLT - University of Venice, Venice, Italy

ARTICLE INFO

Article history:
Received 10 October 2018
Revised 13 May 2019
Accepted 27 February 2020
Available online 29 February 2020

Keywords:
Biclustering
Dominant set
Replicator dynamics
Prior knowledge

ABSTRACT

Biclustering can be defined as the simultaneous clustering of rows and columns in a data matrix and it has been recently applied to many scientific scenarios such as bioinformatics, text analysis and computer vision to name a few. In this paper we propose a novel biclustering approach, that is based on the concept of *dominant-set* clustering and extends such algorithm to the biclustering problem. In more detail, we propose a novel encoding of the biclustering problem as a graph so to use the dominant set concept to analyse rows and columns simultaneously. Moreover, we extend the Dominant Set Biclustering approach to facilitate the insertion of prior knowledge that may be available on the domain. We evaluated the proposed approach on a synthetic benchmark and on two computer vision tasks: *multiple structure recovery* and *region-based correspondence*. The empirical evaluation shows that the method achieves promising results that are comparable to the state-of-the-art and that outperforms competitors in various cases.

© 2020 Elsevier Ltd. All rights reserved.

1. Introduction

Biclustering[1] is usually defined as the simultaneous clustering of both rows and columns of a given data matrix [1–4]. Given a data matrix, the goal of biclustering techniques is to extract subsets of rows that exhibit a “similar” behaviour in a subsets of columns (and vice versa). A key difference between biclustering and clustering is the exploitation of *local information* (instead of global) *to retrieve coherent submatrices*: when performing clustering, data-points are grouped together considering the whole set of features (i.e., we consider global information), in contrast when performing biclustering data-points can be grouped together because they share a coherent behaviour on sub-sets of features (i.e., we consider local information). Bi-clustering was first devised to analyse the expression of genes in microarray data [2,5]. However, recently it has been used in a wide variety of applications ranging from clickstream data analysis [6], to recommender systems [7] and computer vision (e.g., facial expression recognition [8], motion and plane estimation [9,10] and region based correspondences [11]).

The relevant literature on biclustering offers a wealth of techniques[2] that focus on different aspects such as efficiency of the biclustering procedures, interpretability of the biclusters computed by the approach and so forth. Several of such techniques take inspiration from clustering methods and adapt them to biclustering, for example by iteratively performing clustering on rows and columns [13,14].

Following this research line this paper proposes a new biclustering algorithm that is based on the *dominant-set* clustering method. The notion of dominant set is present in several areas of research, ranging from optimization theory to graph theory, game theory and pattern recognition. Taking the clustering viewpoint our input is a set of object V and we wan to group such objects. In this context, a dominant set $C \subseteq V$ is a subset of objects that meets two properties: i) all elements belonging to C should be highly similar to each other, and ii) C can not be a proper subset of any larger cluster. As a consequence, C can be considered a *maximally coherent* set of data items [15,16]. Considering this perspective, a dominant set C is encoded by a characteristic vector $\mathbf{x}$ where an entry x_i represents the likelihood that the object v_i belongs to the extracted cluster. A clustering algorithm based on dominant set is presented in [15,16]. Such algorithm is based on solid theory and has been thoroughly evaluated in different settings. In contrast to standard clustering approaches, dominant set clustering does not partition the data and can be very effective when there is a high level of noise or several outliers may be present. Moreover, dominant-set clustering does not require the similarity matrix to be symmetric. These two features define a strong relationship between dominant-set clustering and biclustering approaches: most biclustering methods do not partition the data and operate on non-squared (and hence not symmetric) data matrices [1].

[*] Corresponding author.
E-mail address: matteo.denitto@univr.it (M. Denitto).

[1] Biclustering is sometimes also called co-clustering in the relevant literature.

[2] We refer to [1,3,5,12] for detailed reviews.

https://doi.org/10.1016/j.patcog.2020.107318
0031-3203/© 2020 Elsevier Ltd. All rights reserved.

A first step toward the usage of dominant sets in the biclustering scenario has been presented in [17]. Authors propose an iterative procedure that retrieves biclusters by shifting and sorting the columns/rows of the input matrix. The dominant set characteristic vector is used to sort and shift the rows and columns. A key point is that in this approach rows and columns are not grouped simultaneously, while the possibility to simultaneously form groups of rows and columns is a crucial element for several biclustering approaches and also for this paper. In a similar way, the approach proposed in [18] addresses the biclustering problem by retrieving the so called *bi-cliques* on a bipartite-graph adjacency matrix. While such technique shares some ideas with our approach, a key difference is that the technique proposed in [18] does not retrieve dominant sets as a result. For this reason we do not discuss this approach in further details.

We believe that it is interesting and important to further investigate how dominant sets can be used in the biclustering scenario. An important stream of work in the biclustering literature formalizes the biclustering problem as an edge-cutting problem in a weighted bipartite graph. The bipartite graph is composed of a set of nodes that represents rows while the other represents the columns [19,20]. However, we can not directly apply the concept of dominant set to a bipartite graph to perform biclustering: a dominant set is equivalent to *maximal clique* [15], and a maximal clique in a bipartite graph is composed by only two nodes. We thus propose to represent the biclustering problem adopting a novel graph representation where entries of the data matrix represent a similarity measure between rows-columns couples. The intuition behind this proposal considers that in various biclustering scenarios (e.g., gene expression, click stream data and recommender systems) entries of the data matrix encode “the importance” of a row for a specific column. Moreover, we modify the bipartite graph adjacency matrix following the ideas proposed in [16,21] so to obtain a theoretically solid dominant set. Finally, we show how to include prior knowledge in the bi-clustering task with a simple principled extension. Prior knowledge in the context of bi-clustering relates row-row and column-column entities of the problem. These additional relations (constraints) allow the user to drive the final solution to include or exclude certain rows or columns. We evaluate the proposed approach both on synthetic and real datasets. Our results show that the approach favourably compares with the state-of-the-art.[3]

The rest of the paper is organized as follows: Section 2 introduces the dominant-set clustering approach; Section 3 describes our algorithm in its basic version. Section 4 describes the extension of our algorithm to include prior knowledge. Each of these two sections contains an experimental evaluation for the proposed approach. Finally Section 6 concludes the paper.

2. Dominant set clustering

The approach proposed in this manuscript extends the dominant set (DS) clustering algorithm [23], a recent and powerful clustering approach, to the biclustering scenario. Before going into the details of the proposed biclustering approach, in this section we provide the necessary background knowledge concerning the dominant set (DS) algorithm – for all the details, we refer interested readers to the recent survey published in [15].

The Dominant Set (DS) algorithm is a graph-based clustering method, in which the data to be clustered is embedded into a graph, where the nodes represent the objects to be clustered, and the weighted edges represent the pairwise similarities between them. The goal is to partition (cluster) the nodes of a graph into disjoint and highly compact sets, which, in the DS algorithm, are represented by the so called Dominant Sets. More in detail, the DS algorithm generalizes the notion of maximal/maximum clique to edge-weighted graphs: searching for a dominant set [23] corresponds in finding a maximal clique in a weighted graph, which turns out to be equivalent in reaching an equilibrium conditions in a non-cooperative game and finding a local solution of a quadratic assignment problem.

More formally, in the DS algorithm, a dataset is embedded into an undirected edge-weighted graph $G = (V, E, \omega)$ with no self loops, in which the vertices V are the items of the dataset. The edges $E \subseteq V \times V$ correspond to the pairwise relations between nodes, and the weight function $\omega : E \to \mathbb{R}_{>=0}$ quantifies the pairwise similarities. The graph, as usual, is represented in terms of an $n \times n$ pairwise symmetric matrix $A = (a_{ij})$ where n is the number of vertices (objects in the dataset):

$$a_{ij} = \begin{cases} w(i,j) & \text{if}(i,j) \in E \\ 0 & \text{otherwise.} \end{cases}$$

Two desirable properties shall hold: having a high *intra-cluster* homogeneity while having a low *inter-cluster homogeneity*. These two properties are fundamental to separate and group objects in the best way possible. Both properties are directly reflected in the combinatorial formulation of the DS (see [23] for the details). In fact, thanks to its one-to-one correspondence with maximal clique, the DS method is able to find compact (highly similar subset of objects) and well separated structures (highly dissimilar).

As stated above, a DS can be found optimizing a standard quadratic assignment problem, defined as:

$$\begin{aligned} &\text{maximize} \quad \mathbf{x}^T A \mathbf{x} && (1) \\ &\text{subject to} \quad \mathbf{x} \in \triangle^n \end{aligned}$$

where A is the similarity matrix of the graph and $\mathbf{x}$ is the so-called *characteristic vector* in which each ith component represents the probability of belonging to a dominant set. The vector $\mathbf{x}$ lies in the n-dimensional simplex $\triangle^n$, i.e. $\sum_i \mathbf{x}_i = 1, \forall i \ , x_i \geq 0$. It has been shown in [15] that, if $\mathbf{x}$ is a strict local solution of (1) then its support $\sigma(\mathbf{x}) = \{i \in V | x_i > 0\}$ is a dominant set. A DS is therefore found by looking at a local solution of (1). A way to find such local optimizer, is to use a result from the evolutionary game theory [24] known as *replicator dynamics* (RD). This approach considers a scenario whereby individuals are repeatedly drawn at random from a large, ideally infinite, population to play a two-player game. In contrast to classical game theory, here players are not supposed to behave rationally or to have complete knowledge of the details of the game. They act instead according to an inherited behavioural pattern, or pure strategy, and an evolutionary selection process operates over time on the distribution of behaviours. In particular, we adopt the iterative discrete-time replicator dynamics, which are defined in Eq. (2):

$$x_i(t+1) = x_i(t) \frac{(A\mathbf{x}(t))_i}{\mathbf{x}(t)^T A \mathbf{x}(t)} \qquad (2)$$

At convergence of Eq. (2) certain components of x will emerge ($x_i > 0$) while others get extinct ($x_i = 0$). The convergence of Eq. (2) can be reached by fixing a maximum number of steps or when the distance between two successive iterations is smaller than ϵ ($||(\mathbf{x}(t) - \mathbf{x}(t+1))||_2 \leq \epsilon$).

Summarizing, to cluster a dataset, the dominant set algorithm follows these steps:

[3] A preliminary version of this paper appeared in [22]: the current manuscript contributes upon [22] in the following directions: *i)* from a methodological perspective we proposed a principled extension of Denitto et al. [22] showing how to include prior knowledge in the DSB framework, grounding this finding in the theory of Dominant Set; and *ii)* we assessed the goodness of the proposed method *with and without* prior knowledge on an extensive novel experimental session.

1. Construct an undirected graph $G = (V, E, \omega)$ with no self-loop in which V is the set of observations in the dataset, E is the set of pairwise relations and ω is a pairwise similarity between all the vertices. The similarity ω can be given in input or can be computed using any similarity function between objects (e.g. the cosine similarity for vectors).
2. Store the graph in an $n \times n$ matrix A $(n = |V|)$.
3. Initialize $\mathbf{x}$ in the barycenter of the simplex $(x_i = 1/n \ \ \forall i = 1 \ldots n)$
4. Run the replicator dynamics (Eq. (2)) until convergence
5. Extract the support from $\mathbf{x}$. The set of nodes in the support corresponds to a dominant set.
6. A dominant set corresponds to a single cluster: if the problem requires to find more clusters (or if a complete partitioning is necessary), remove the extracted nodes from the graph (mask/remove the corresponding rows and cols in the matrix A) and iterate again on the remaining ones.

3. The proposed approach: biclustering with dominant sets

In this Section we introduce our extension of the Dominant Sets algorithm to the biclustering case. As discussed in Section 1, the goal of biclustering is to cluster simultaneously the columns and the rows of a given data matrix. We denote the data matrix as $D \in \mathbb{R}^{n \times m}$, and we indicate with $R = \{1, \ldots, n\}$ and $K = \{1, \ldots, m\}$ the set of row and column indices. We use D_{TL}, where $T \subseteq R$ and $L \subseteq K$, to represent the submatrix with the subset of rows in T and the subset of columns in L. We can now define a *bicluster* to be a submatrix D_{TL}, where the subset of rows of D with indices in T exhibits a “coherent behaviour” (in some sense) across the set of columns with indices in L, and vice versa. The definition of the coherence criterion impacts on the biclusters that we want to retrieve (for a comprehensive survey of biclustering criteria, see [1,25]).

In this paper we propose to tackle biclustering by exploiting the principles of dominant sets. As mentioned in Section 1 a preliminary approach toward this objective was proposed in the literature. Specifically, authors of [17] propose an algorithm that clusters rows and columns iteratively but does not fully exploit the potentials of dominant sets. In more detail, the approach described in [17] proposes a weighted correlation measure that defines a similarity matrix between the rows of the given data matrix. Dominant-set clustering is then applied to such data matrix, exploiting the characteristic vector $\mathbf{x}^C$ to sort the rows of the matrix. As a result of this procedure, rows that belong to the bicluster are placed at the bottom of the data matrix. After this step, authors compute a similarity matrix for the columns weighting the correlation by using $\mathbf{x}^C$. This results in higher weights for rows that belong to the bicluster. Next, authors apply dominant-set clustering to the columns similarity matrix. The idea is that weighting the columns correlation by using the characteristic vector (computed on the rows) should help in extracting a subset of columns that exhibit a coherent behaviour in that particular subset of rows. Finally, columns are ordered according to their characteristic vector and the entire process is iteratively repeated twice for rows and columns [17]. The resulting data matrix now contains the bicluster in the bottom-right position. To extract the bicluster from the matrix, authors compute the correlation between consecutive rows (starting from the bottom), and they stop when such correlation is below a certain threshold (same procedure applies for retrieving bicluster columns). In summary, the work proposed in [17] exploits the output of dominant-set clustering to order rows and columns iteratively so to position the bicluster in the bottom-right portion of the data matrix.

Other techniques consider the use of a weighted bipartite graph representation to address biclustering. Usually, such techniques represent the set of rows R and the set of columns K with two distinct sets of nodes. Then, the approaches connect only nodes belonging to different sets weighting the edges with the data matrix entries. The biclustering problem can now be cast as an edge cutting problem where the remaining edges represent the rows and the columns that belong to the bicluster [20]. The cut is performed by considering a pre-defined objective function.

Our proposed algorithm exploits the underlying DS theory hence requires to represent the input objects with a similarity graph: our idea is to create a graph where the nodes represent the rows *and* the columns, i.e. a graph with $(n + m)$ nodes (for an input data matrix with n rows and m columns): a subset of nodes can therefore represent a subset of rows, a subset of columns, or a subset of rows *and* columns – this last being a possible bicluster. Being more formal, our biclustering problem is represented by a graph $G = (V, E, \omega)$ where vertices $V = \{1, \ldots, n + m\}$ represent the union of rows $(\{v_1, \ldots, v_n\})$ and columns $(\{v_{n+1}, \ldots, v_{n+m}\})$ of a data matrix D. In order to get only subsets of rows *and* columns (i.e. biclusters), we have to force a structure like a bipartite graph. To get this, we first set the portion of A that represents the similarities for row-row and column-column pairs to 0 – hence forcing no connections between such vertices (we will see in Section 4 how this part, if not set to zero, can be fruitfully used to encode a-priori knowledge). Second, we insert the data matrix D in a particular position of the adjacency matrix A, i.e. in the set of edges connecting the two partitions of the bipartite graph (the rows and the columns): the idea is to set the similarity between a row i and a column j as the entry D_{ij} of the input data matrix. This choice is reasonable for many biclustering problems, especially those which involve the analysis of preference/consensus matrices [26]. In such contexts, the entry D_{ij} of the matrix represents how much the row i (object i) prefers a given column j (feature j): in [26] authors show that several problems can be interpreted from this perspective. In more detail, we set $A([1, \ldots, n], [n + 1, \ldots, n + m]) = D$ – again, this represents the set of edges connecting the two partitions of the bipartite graph. Then, to have a symmetric adjacency matrix, we also set $A([n + 1, \ldots, n + m], [1, \ldots, n]) = D^T$. This is mandatory for the dominant set framework because it requires a square pairwise similarity matrix.[4]

We now have a squared similarity matrix that represents a bipartite graph, hence we can apply dominant set clustering approaches to such a matrix and directly obtain a bicluster. In more detail, the adjacency matrix we defined has high values[5] only in positions that encode row-columns relationships. Hence, the dominant set should be a group of rows that have high similarities in a subset of columns (and vice versa). However, recall that, as mentioned in Section 2, from graph theory we know that a dominant set of A is equivalent to a maximal clique in the correspondent graph. Hence, since we encode our problem by using a bipartite graph (as we set row-row/column-column entries to be zero), a maximal clique will always have just two vertices: one row and one column. In particular, we can not find a maximal clique with three nodes, because there will certainly be a missing edge: actually, a set of three nodes will contain at least two nodes of the same type (rows or columns), and by construction there are no edges between row-row or column-column. To avoid this problem, we insert a negative value $-\alpha$ (where $\alpha \geq 0$) on the main diagonal of the similarity matrix A. This is equivalent to solve a standard quadratic problem where we increase the values of the off-diagonal entries of A by α and we set the main diagonal to 0 [16,27]. Crucially, by inserting α on the off-diagonal entries we introduce row-row and column-column edges, hence obtaining a

[4] Please note that this representation has been also used in another work to match set of objects [21] in k-partite graph-

[5] Without loss of generality we assume a positive data matrix D.

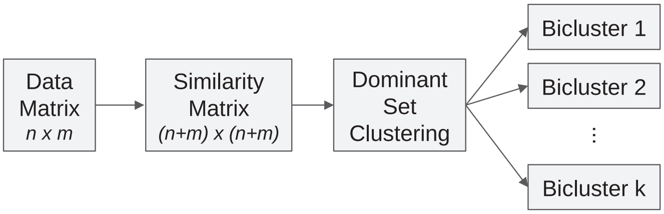


Fig. 1. Flowchart of the method. In *italics* the sizes of the matrices involved on each step.

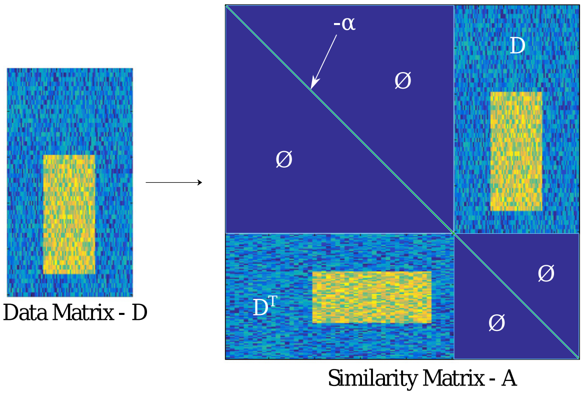


Fig. 2. Procedure for building the similarity matrix.

classic graph (not bipartite). Hence when we compute dominant sets on this version of A we will obtain maximal cliques, where a subset of rows will be selected simultaneously with a subset of columns. The main idea of the proposed algorithm is graphically sketched in Figs. 1 and 2 provides a visual representation of the procedure that we use to build the similarity matrix A.

Intuitively, this is possible because, regardless of the value used for α, the information we care about is preserved in the rows-columns portions of A (since we increase all entries by α and hence it is not informative). It has also been theoretically proved that increasing the value of α will increase the dimension of the resulting clique [15,16,27], as shown in Fig. 3.

In summary, given a data matrix D with n rows and m columns, our approach for biclustering with dominant sets – named *Dominant Set Biclustering (DSB)* – performs the following steps:

1. we encode the biclustering problem by considering a graph with $n+m$ vertices. The first n vertices represent the rows while the remaining m vertices represent the columns.
2. We then connect rows and columns vertices with weights that correspond to the entries of D. All other connections (row-row and column-column) are set to zero.
3. We set to $-\alpha$ the weights of self connections.
4. The resulting similarity matrix A, of dimension $(n+m) \times (n+m)$, can now be given in input to the dominant sets algorithm: a dominant set in such graph is a maximal clique that identifies a group of rows that exhibit high similarities with a group of columns, hence a bicluster.

The algorithm requires two parameters which are i) α (i.e., the value for diagonal weights), ii) the convergence criterion for the replicator dynamics. The setting of α is definitely important, and is typically application dependent (as shown in the experimental section). In Section 5 we present some guidelines and comments on the impact of this parameter. On the contrary, the convergence criterion does not impact too much the results, and can be defined either by specifying a maximum number of iterations or by setting a threshold between consecutive changes of $\mathbf{x}$. Notice that this algorithm can recover only a single bicluster at a time. However, as often employed in the literature [2,28,29]), we can retrieve several biclusters by "masking" the obtained bicluster and then search for the next one. In our approach we decided to mask the extracted bicluster by inserting zeros in the corresponding positions of the adjacency matrix A.

3.1. Experimental evaluation

We evaluate the basic approach by considering two sets of synthetic datasets and one Computer Vision dataset (namely Multiple Structure Recovery) divided in two problems (motion and planar segmentation).

3.1.1. Synthetic experiments

The two synthetic benchmarks simulate gene expression matrices that contain a single bicluster. In the first dataset, we implanted biclusters with constant value (we name this "Constant Bicluster Benchmark"), while in the second dataset we use additively coherent biclusters (we name this "Evolutionary Bicluster Benchmark").

We use the following procedure to generate the matrices in both cases: i) we generate a 50 × 50 matrix that contains random values, uniformly distributed between 0 and 1; ii) we insert a constant valued (or additively coherent valued) bicluster. The dimension of such bicluster is 25% of the matrix size and it is inserted in a random position of the data matrix; iii) finally, we perturb the entire matrix using Gaussian noise. The standard deviation of the Gaussian noise is a percentage of the difference between the mean of the entries belonging to the bicluster and the mean of the background. We consider 5 different noise levels (i.e. percentages) that

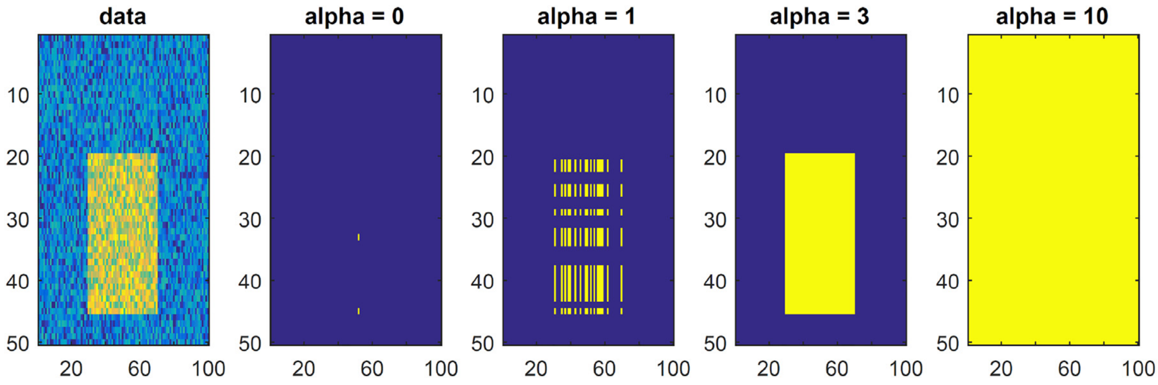
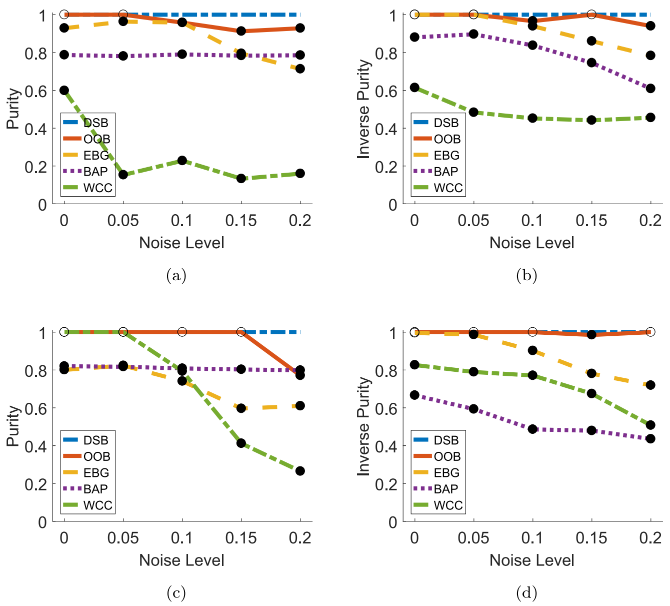


Fig. 3. Different results varying alphas.



(a) (b) (c) (d)

Fig. 4. Purity (Fig. 4a,c) and Inverse Purity (Fig. 4b,d) for matrices with constant (Fig. 4a,b) and additive coherent (Fig. 4c,d) biclusters

range from 0 (no noise) to 0.2 (high noise). We generate 30 matrices for each noise level (75 matrices in total).

We assess the quality of the retrieved biclusters by using two standard indices, also used in previous work [30]: i) *purity*: defined as the percentage of points that are retrieved by the algorithms which belong to the real bicluster; ii) *inverse purity*: defined as the percentage of points belonging to the real bicluster which were retrieved by the algorithms. Formally, the indices are calculated as follows:

$$\text{Purity} = \frac{|C \cap L|}{|C|}, \qquad \text{Inverse Purity} = \frac{|L \cap C|}{|L|};$$

where C is the bicluster found by the algorithm and L is the real bicluster.

In all the experiments we set the convergence threshold for the Dominant Set Biclustering (DSB) as 10^{-20}, whereas α was set to 3 (for some comments and guidelines on how to set this parameter please see Section 4). We compare the proposed approach with four biclustering algorithms, including the previous one adopting dominant set (mentioned in Section 1, which we refer to as Weighted Correlation Coefficient - WCC). For the OOB, EBG and BAP algorithms we consider the results published in [9], while for WCC we implemented the approach following the indications presented in [17] and we use the values suggested in that paper to tune the parameters.

Fig. 4 reports the results achieved for the Constant and Evolutionary Bicluster benchmarks. Each graph reports the value of purity (Fig. 4a,c) and inverse purity (Fig. 4b,d) for all the tested methods, varying the noise level. Each point reports the performance value averaged over 30 runs with the specified noise level. If the difference between the different methods and our proposed approach is statistically significant[6] we use a full marker.

Results clearly show that our approach significantly outperforms the competitors, and this is particularly true when considering increasing noise levels. This confirms that dominant sets hold great potentials in retrieving biclusters in situations that exhibit a high level of noise. Notice that the weighted correlation coefficient impacts on the performance for WCC, which is indeed expected. The motivation is the follow: if we can select the correct columns that are involved in the bicluster, then the behaviour of the bicluster (in the selected columns) will be similar to the background because the value of the bicluster is constant. Hence, it is difficult for this method to differentiate between these two situations. This also explains the better performance of WCC in the evolutionary bicluster benchmark: in this case the background and the bicluster have different behaviours (because the background is constant while the bicluster evolves). In any case, the proposed approach is superior in both situations, indicating that the use of a solid framework to encode dominant sets provides significant benefits.

3.1.2. Multiple structure recovery

Multiple structure recovery (MSR) relates to the extraction of multiple models from noisy or outlier-contaminated data. MSR is a challenging and significant problem, which is a crucial element for many computer vision applications [32,33,31]. In general, an instance of an MSR problem can be represented by a *preference matrix* that contains the points under analysis, along one dimension, and the hypotheses/structures for the models, in the other dimension. The entry (i, j) in this matrix indicates how well a given point i is represented by the hypothesis/structure j.

We focus our analysis on the Adelaide dataset, which was previously used to assess the quality of biclustering algorithms [9]. Such dataset involves two type of MSR problems: motion and plane estimation. Motion segmentation takes as input two different images of the same scene, where several objects move independently, and the goal is to recover subsets of point matches that undergo the same motion. Plane segmentation takes two uncalibrated views of a scene, and aims at retrieving the multi-planar structures by fitting homographies to point correspondences. The AdelaideRMF dataset[7] comprises 38 image pairs (19 for motion segmentation and 19 for plane segmentation), with matching points contaminated by strong outliers. The dataset also offers the ground-truth segmentations. Following [9,34], we also adopt the misclassification errors to assess the results.

Tables 1 and 2 show the achieved results. For what concerns the proposed approach, due to the heterogeneity of the dataset (38 matrices, which characteristics are drastically different – see the first two columns), we tested the values of α in the set $[3, 1, 0.1, 10^{-5}, 10^{-7}]$ (the convergence threshold for replicator dynamics was set to 10^{-5}). Then we reported two different results, in the last two columns of the tables. The last column (*DSB best*) reports the results for the DSB algorithm that we achieved when we consider the best performance with respect to the misclassification error for each different matrix (varying the parameters). The results in the sixth column (*DSB best set*), reports the results achieved by selecting a single set of parameters for each dataset (one for the motion segmentation and one for the plane estimation). While the results in the last column are slightly worse the difference is small, demonstrating that dominant sets are robust to noise and the presence of outliers. We compare our approach to recent state of the art methods: T-linkage [35], RPA (Robust Principal Analysis [36]) and RCMSA (Random Cluster Model Simulated Annealing [37]). When considering these other techniques, our approach improves the results in the plane segmentation dataset Table 2, and provides comparable results on the motion segmentation dataset Table 1.

4. Injecting prior knowledge in dominant set biclustering

In this section we describe an extension of the DSB which considers the exploitation of a priori information. Even if this

[6] Statistical significance was measured by performing a t-test for each noise level (on the result of the 30 matrices), the significance level was set to 5%.

[7] https://cs.adelaide.edu.au/~hwong/doku.php?id=data.

Table 1
Misclassification error (ME %) for motion segmentation. k is the number of models and % out is the percentage of outliers.

	k	%out	T-lnkg	RCMSA	RPA	DSB best set	DSB best
biscuitbookbox	3	37.21	3.10	16.92	3.88	10.42	6.17
breadcartoychips	4	35.20	14.29	25.69	7.50	5.48	5.48
breadcubechips	3	35.22	3.48	8.12	5.07	5.21	5.21
breadtoycar	3	34.15	9.15	18.29	7.52	11.44	11.44
carchipscube	3	36.59	4.27	18.90	6.50	4.24	4.24
cubebreadtoychips	4	28.03	9.24	13.27	4.99	9.48	9.48
dinobooks	3	44.54	20.94	23.50	15.14	14.16	14.16
toycubecar	3	36.36	15.66	13.81	9.43	16.00	16.00
biscuit	1	57.68	16.93	14.00	1.15	16.36	16.36
biscuitbook	2	47.51	3.23	8.41	3.23	2.63	2.63
boardgame	1	42.48	21.43	19.80	11.65	8.96	8.96
book	1	44.32	3.24	4.32	2.88	10.69	10.69
breadcube	2	32.19	19.31	9.87	4.58	11.57	9.50
breadtoy	2	37.41	5.40	3.96	2.76	3.12	3.12
cube	1	69.49	7.80	8.14	3.28	3.31	3.31
cubetoy	2	41.42	3.77	5.86	4.04	4.81	4.81
game	1	73.48	1.30	5.07	3.62	1.71	1.71
gamebiscuit	2	51.54	9.26	9.37	2.57	4.57	4.57
cubechips	2	51.62	6.14	7.70	4.57	7.04	7.04
mean			9.36	12.37	5.49	7.96	7.62
median			7.80	9.87	4.57	7.04	6.17

Table 2
Misclassification error (ME %) for planar segmentation. k is the number of models and % out is the percentage of outliers.

	k	%out	T-lnkg	RCMSA	RPA	DSB best set	DSB best
unionhouse	5	18.78	48.99	2.64	10.87	25.00	25.00
bonython	1	75.13	11.92	17.79	15.89	4.04	4.04
physics	1	46.60	29.13	48.87	0.00	2.83	0.94
elderhalla	2	60.75	10.75	29.28	0.93	5.14	2.80
ladysymon	2	33.48	24.67	39.50	24.67	10.54	10.54
library	2	56.13	24.53	40.72	31.29	13.95	13.95
nese	2	30.29	7.05	46.34	0.83	0	0
sene	2	44.49	7.63	20.20	0.42	0.40	0
napiera	2	64.73	28.08	31.16	9.25	13.24	13.24
hartley	2	62.22	21.90	37.78	17.78	3.12	1.56
oldclassicswing	2	32.23	20.66	21.30	25.25	8.44	8.44
barrsmith	2	69.79	49.79	20.14	36.31	51.03	51.03
neem	3	37.83	25.65	41.45	19.86	25.72	15.76
elderhallb	3	49.80	31.02	35.78	17.82	25.88	18.82
napierb	3	37.13	13.50	29.40	31.22	20.84	20.84
johnsona	4	21.25	34.28	36.73	10.76	20.37	20.37
johnsonb	7	12.02	24.04	16.46	26.76	19.87	19.87
unihouse	5	18.78	33.13	2.56	5.21	3.69	3.69
bonhall	6	6.43	21.84	19.69	41.67	38.76	38.76
mean			24.66	28.30	17.20	15.41	14.19
median			23.38	29.40	17.53	13.24	13.24

exploitation is not new in both contexts of clustering and biclustering [11,38], its consideration in the context of the dominant set theory is completely novel. A common assumption [11,38] is to consider pair-wise information to guide the grouping, typically to *encourage* points to be grouped together. In our case, however, the type of information that we can consider is double: i) we can *favour* rows (or columns) to be grouped together and ii) we can *prevent* rows (or columns) to be grouped together. Injecting such a priori information into our framework is very straightforward. Actually, remember that putting $-\alpha$ on the main diagonal has the effect of assigning to each row-row and column-column similarity the exact same value (thus not informative) α. Looking at it as an adjacency matrix, this means adding edges between row nodes (and between columns nodes) with the same weight. However, when a priori knowledge is available we could modify such values in order to favour or not particular couples of rows/columns to be grouped together. Moreover, removing particular edges we can prevent rows/columns to be grouped together. Thus, by simply modifying the matrix A we can easily integrate a priori information. Another feature of the designed similarity matrix is the possibility of handling information deriving from different sources at once. In fact, we can integrate the similarity between rows and columns with some other knowledge concerning only rows (or only columns). More in detail, the matrix A becomes:

1. $A \in \mathcal{R}^{(n+m)\times(n+m)}$;
2. $A([1,\ldots,n],[n+1,\ldots,n+m]) = D$;
3. $A([n+1,\ldots,n+m],[1,\ldots,n]) = D^T$;
4. $A(i,i) = -\alpha,\ \forall i \in [1,\ldots,n+m]$;
5. $A([1,\ldots,n][1,\ldots,n]) = R$;
6. $A([n+1,\ldots,n+m][n+1,\ldots,n+m]) = C$;

where R is a matrix composed of the prior knowledge on the rows and C is a matrix containing the prior knowledge on the columns. In Fig. 5 we show how the similarity matrix A is modified for this task. A critical aspect here, is to balance the different components of the matrix such that the weights do not differ too much.

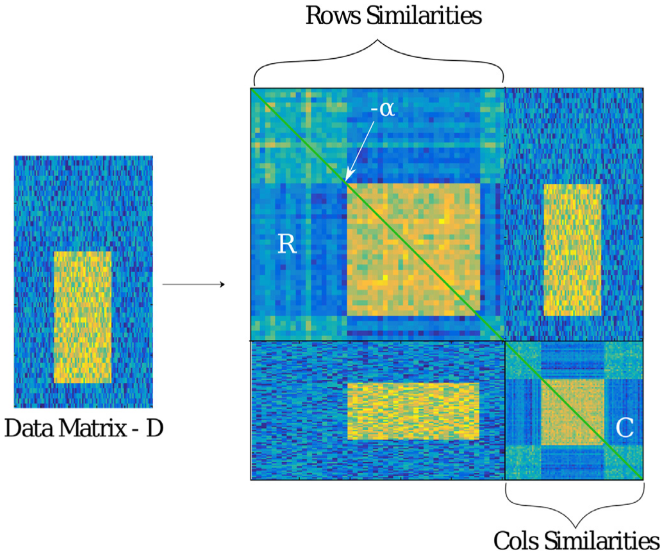


Fig. 5. Procedure for injecting prior Knowledge in the similarity matrix.

This is particularly important in order to not favour uninformative/trivial solutions. For example if the row-row weights (top-left block of the matrix) result unbalanced w.r.t. the row-column the final solution will be mostly comprised by row elements which is not the aim of this work. Balancing the similarity matrices is then nothing than an easy task because, most of the time, the a-priori knowledge comes or are computed with a different function then the one used in matrix D, hence the values have different bound or meaning. In order to do so, we introduced a parameter β which is used to scale the values in correspondence of the row-row/column-column similarities. This scaling makes the entire matrix in the same range. The parameter α is kept as in the standard DSB version (see Section 3) in order to find a solution wider as needed. The final matrix A is then the following:

1. $A \in \mathcal{R}^{(n+m)\times(n+m)}$;
2. $A([1,\dots,n],[n+1,\dots,n+m]) = D$;
3. $A([n+1,\dots,n+m],[1,\dots,n]) = D^T$;
4. $A(i,i) = -\alpha,\ \forall i \in [1,\dots,n+m]$;
5. $A([1,\dots,n][1,\dots,n]) = \beta * R$;
6. $A([n+1,\dots,n+m][n+1,\dots,n+m]) = \beta * C$;

Please note that the replicator dynamics (see Eq. (2)) are not affected by this configuration change, and thus the procedure to obtain biclusters from A is the same described in Section 3.

4.1. Experimental evaluation

A recent Computer Vision scenario where biclustering with prior knowledge already showed its potential is the one called *region-based correspondence* (RBC) [11,39]. Generalizing, in RBC the problem is formulated as that of finding regions on two different shapes that behave similarly and can thus be easily put in correspondence. This problem is different from another well-studied task, called *shape co-segmentation*, since in RBC the goal is **not** to find meaningful semantic segments in various shapes (*e.g.*, limbs in animal shapes), but rather to determine regions in the two shapes that are in correspondence [11].

As presented in [39] and [11], RBC can be tackled as a biclustering problem. In fact, following the framework in the cited papers, we can obtain an affinity matrix representing the similarities between different shape vertices. Analysing such matrix with a biclustering algorithm we would obtain a subset of vertices of the first shape (rows) behaving coherently in a subset of vertices of the second shape (columns), which is the goal of RBC. Concerning RBC-specific techniques, the most relevant are represented by the recent *stable region correspondences* (SRC) approach [39] and by the S^{4}B technique [11]. Concerning these techniques, it is important to highlight that SRC does not involve any a priori information, whereas S^{4}B exploits geodesic distances to encourage near points to be grouped together.

The experimental evaluation is based on FAUST [40], a challenging recent dataset containing 100 scanned human shapes (10 poses of 10 subjects). This dataset presents both near-isometric (different poses of the same subject) and non-isometric deformations (due to the significant variability between different subjects). All of the shapes have the same number of vertices, and the ground-truth one-to-one correspondence (or map) between each pair of shapes is available. We analyse the same 50 randomly selected pairs of shapes in [11], which are divided in three scenarios: Scenario1, which contains pairs of shapes of the same subject in different poses, Scenario2, containing pairs of different subjects in the same pose, and the most challenging Scenario3, including pairs of different subjects in different poses. The accuracy of a given algorithm can be assessed by comparing the extracted regions with the ground-truth mapping between the two shapes. We measure the accuracy of the results as proposed in [11], the higher the better.

On the described task we evaluate the proposed DSB algorithm without and with prior information. We extract 20 biclusters for each couple varying the parameters $\alpha \in \{10^{-5}, 30, 60\}$ and $\beta \in \{0, 8, 12\}$. Once the 20 biclusters have been extracted, since the evaluation method expects one single label for each vertex (no overlap between biclusters), we assign to each point the label of the first bicluster including it. We then consider the best set of biclusters for each couple of shape. Regarding the prior information, we first compute the geodesic distance between all points belonging to the first shape, repeating the process for the second shape. This gives us two distance matrices D_1 and D_2 of size $n \times n$ and $m \times m$ respectively. Both matrices are then turned to similarities with a linear transformation, hence:

$$R = \max(D_1) - D_1$$

$$C = \max(D_2) - D_2$$

then we used directly those matrices in the affinity matrix of the DSB (see Section 4) with the parameter α and scaling constant β.

To provide a first idea on how the proposed framework works in this context, we reported in Fig. 6 some results: we can observe that, qualitatively, the results on the 3 scenarios from FAUST are particularly good.

In a first set of experiments, we compare the proposed DSB (with and without prior knowledge) with SRC and S^{4}B. We also include in the comparison the SSBi technique [10], an algorithm similar to the algorithm of S^{4}B, with the only difference that it does not exploit any prior knowledge. The results are reported in Table 3. As shown, DSB comparably performs with respect to the current state of the art. Particularly, we can see that the baseline method DSB provides extremely good results when compared with SRC, SSBi and S^{4}B. When prior information is included into DSB, the performances considerably increase, improving the state of the art. An important consideration to formulate is that the proposed method extracts one bicluster at a time, thus the results can vary significantly on the basis of how biclusters are masked and merged. This strengths even more the comparison with S^{4}B, which extracts many biclusters simultaneously.

As a second set of experiments we tested how DSB (in both variants) works as initialization for other algorithms. Actually, other methods produce point-to-point correspondences based on geometric features of the shapes. For example, a popular point-to-point correspondence method is the so called *blended intrinsic maps* (BIM) [41]. In order to use BIM in the context of

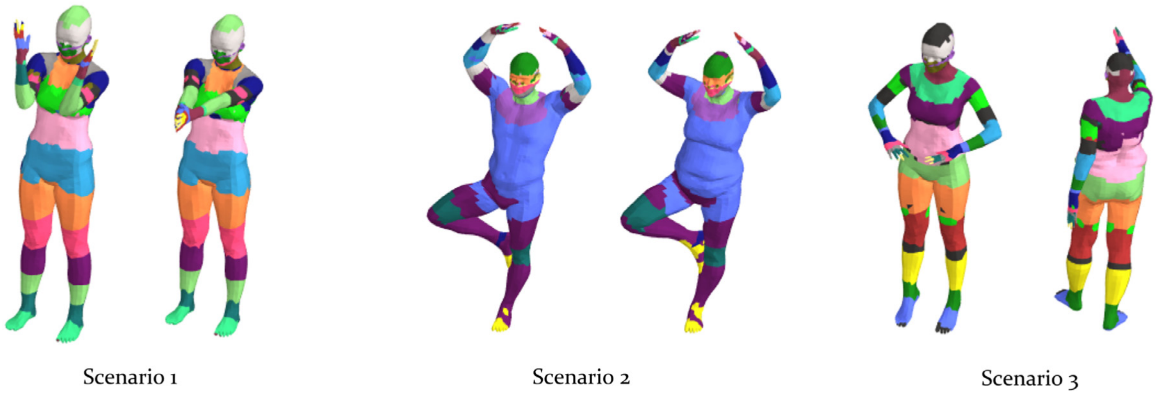


Fig. 6. Qualitative results of our method on three scenarios from FAUST dataset.

Table 3
Results on the FAUST dataset using the different approaches. The top/bottom tables show mean/median scores for each scenario, and the global mean/median score.

Shapes Couples	Stable Region	SSBi	S^4B	DSB best no PK	DSB best w. PK
scenario1	90.6	30.09	95.39	95.36	96.35
scenario2	84.81	26.69	95.08	93.04	96.48
scenario3	86.19	32.72	94.68	89.98	94.66
global	86.58	31.8	94.8	90.93	95.05
Shapes Couples	Stable Region	SSBi	S^4B	DSB best no PK	DSB best w. PK
scenario1	93.23	27.98	96.94	95.09	96.8
scenario2	85.39	29.58	94.42	91.92	97.11
scenario3	87.92	33.67	95.43	89.38	94.89
global	89.33	31.26	95.52	90.12	95.02

Table 4
Results on the FAUST dataset using *BIM* with different initializations. The top/bottom tables show mean/median scores for each scenario, and the global mean/median score.

Shapes Couples	BIM Voronoi	BIM S^4B	BIM DSB best w. PK
scenario1	94.97	96.89	95.97
scenario2	93.74	93.55	96.31
scenario3	92.93	92.79	93.58
global	93.26	93.36	94.14
Shapes Couples	BIM Voronoi	BIM S^4B	BIM DSB best w. PK
scenario1	95.49	97.22	97.7
scenario2	93.16	92.98	96.83
scenario3	92.96	92.6	94.04
global	93.1	93.15	94.5

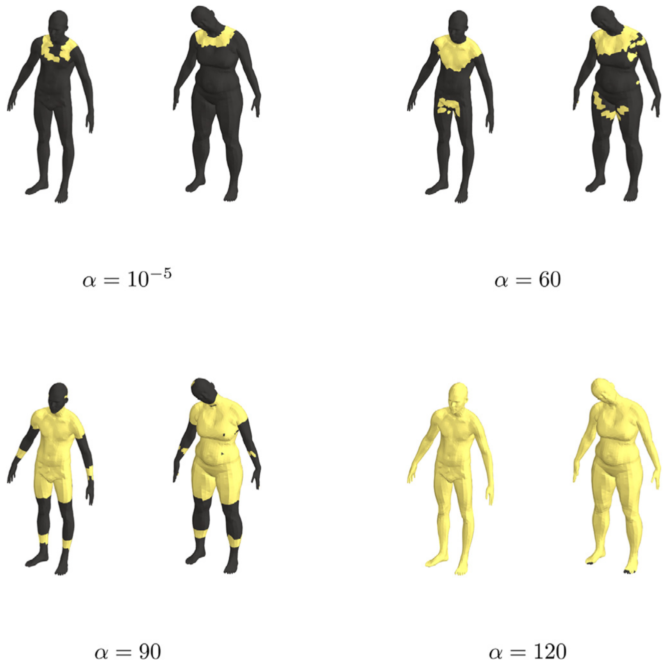
corresponding regions problem, we follow [11,39] and use the point-to-point mapping to transport the segmentation computed on one shape to the other. However, since its performance is highly influenced by the starting segmentation, we evaluate the point-to-point mapping using three possible segmentations: (i) based on geodesic Voronoi cells around a *farthest point sampling* [42], which provides segments of uniform size; (ii) based on the output labels of S^4B (which exploits geodesic distances) and; (iii) based on the output of DSB with prior knowledge. This gives us a starting segmentation, which we transfer to the second shape using the correspondences provided by BIM. Table 4 reports the result of BIM when initialized with the different methods. As can be seen, the regions provided by DSB with prior knowledge are better than those obtained with the competitors in most of the cases.

5. Setting the parameters α and β

The proposed algorithm, in the most complete version (i.e. with also prior knowledge), depends on the parameters α and β: the former represents the value subtracted from the diagonal of the built similarity matrix, whereas the second weights the importance of the a priori knowledge. As common in many clustering and biclustering algorithms, the proper setting of these parameters depends on the given application, and can influence the result.

However, for what concerns α, we can derive some guidelines on how to set its value by exploiting some recent theoretical results derived for Dominant Sets [16,27]. In particular, it has been proved in [27] that the value of α is bounded by the largest eigenvalue of the similarity matrix. In other words, setting α to a value larger than such largest eigenvalue results in a dominant set which covers the whole set of objects. In our context, this means that we obtain a bicluster which covers the whole matrix. Second, in [16] it has been shown that the value of α is linked to the dimension of the obtained dominant set. In our context, this means that larger values of α permit to get larger biclusters; in Fig. 7 we show, for the SRC problem, the extracted regions when varying α: as expected, larger values permit to get larger regions.

The second parameter, β, quantifies the importance of the a priori information: the larger this value, the more important is the prior knowledge. This is a common scenario in pattern recognition (i.e. regularization): the selection of the best parameter in this context represents a still unsolved challenge, typically faced using context-dependent knowledge. In general, in our approach a high β permits to give higher importance to the row-row and column-column relations with respect to the row-column ones. In the SRC problem, this is equivalent to look for more compact regions (i.e. regions with nearby vertices), possibly loosing coherence between the two shapes. For example, in the first row of Fig. 8, we can observe that when increasing β our approach is able to recover regions with less holes. In general, however, changing β may also lead to different regions, as shown in the second row of Fig. 8 (where we used a larger α). This is reasonable, since the

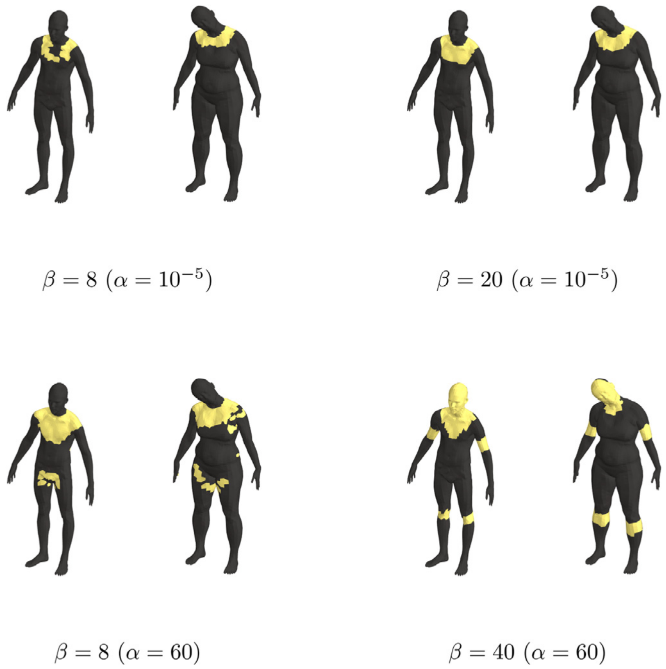

$\alpha = 10^{-5}$

$\alpha = 60$

$\alpha = 90$

$\alpha = 120$

Fig. 7. Influence of parameter α.


$\beta = 8\ (\alpha = 10^{-5})$

$\beta = 20\ (\alpha = 10^{-5})$

$\beta = 8\ (\alpha = 60)$

$\beta = 40\ (\alpha = 60)$

Fig. 8. Influence of parameter β.

final result is obtained through an optimization procedure which considers all ingredients.

Concluding, we can report that in our experiments the selection of the proper values for the parameters α and β was not so complicated, obtaining reasonable results after few trials. Moreover, in the MSR case, results do not change too much when varying these values. This can be seen by looking at the last two columns of Tables 2 and 1; in the last column, we used for each experiment a different set of parameters, whereas in the second-to-last we use the same set: the difference is very narrow.

6. Conclusions

This paper proposes a novel approach to address the biclustering problem. The proposed algorithm extends the definition of dominant sets (which is already widely exploited for clustering) to the biclustering scenario. In more detail, we propose a novel paradigm to represent the biclustering problem that has a sound theoretical basis. The main idea is to embed the bipartite graph – typical of the biclustering scenario – in a standard similarity graph, to be analysed and processed using the dominant sets algorithm. Such novel paradigm allows us to retrieve from the graph Dominant sets, that represent the biclusters, efficiently by using standard discrete-time replicator dynamics. We also proposed a variant of the paradigm that can introduce into our framework the prior knowledge on the relations between row-row and column-column; the possibility of injecting this prior knowledge into the biclustering scenarios can be very useful in many scenarios, such as the analysis of biological data [43]. The whole framework thus results in a flexible and appealing algorithm for a variety of usages. We empirically evaluated the performance of the algorithm on both synthetic and real datasets, involving challenging computer vision applications, such as Multiple Structure Recovery (MSR) and Region-Based correspondence (RBC). Results are encouraging when compared to recent state-of-the-art methods.

From a general perspective, we think that this paper opens the route to the exploitation of classical similarity-based clustering approaches to solve the biclustering problem. More in detail, among other contributions, here we have proposed a scheme which permits to formulate the biclustering problem as a classical clustering problem, which can – in principle – be solved with other pairwise similarity-based methods. In this sense it would be very interesting to investigate how and when alternative schemes can be more adequate that the dominant sets algorithm. More in relation to the proposed scheme, an open problem which still deserves some attention is the proper setting of the parameters α and β. As discussed above, an adequate setting of these values is definitely application-dependent, and can influence the result of the algorithm. However, a possibility is to exploit the recent results obtained in the dominant set field [16,27], where some theorems and bounds have been provided to help the setting of the value of the parameters. Finally, even if we demonstrated the usability of the proposed scheme in two challenging real scenarios, we think it can be also used in other contexts. In particular, we think it would be worth to apply the proposed framework also for biological data analysis, for example in the context of analysis of expression data. In such cases, prior knowledge on relations between genes can be easily derived from biological studies, and can be successfully exploited to improve biclustering results – this has been shown for example for the biclustering approaches based on topic models [44,45].

References

[1] S. Madeira, A. Oliveira, Biclustering algorithms for biological data analysis: a survey, IEEE Trans. Comput. Biol.Bioinf. 1 (2004) 24–44.

[2] Y. Cheng, G. Church, Biclustering of expression data, in: Proc. Eighth Int. Conf. on Intelligent Systems for Molecular Biology (ISMB00), 2000, pp. 93–103.

[3] J. Flores, I. Inza, P. Larranaga, B. Calvo, A new measure for gene expression biclustering based on non-parametric correlation, Comput. Methods. Programs Biomed. 112 (3) (2013) 367–397.

[4] R. Henriques, C. Antunes, S.C. Madeira, A structured view on pattern mining-based biclustering, Pattern Recognit. 48 (12) (2015) 3941–3958.

[5] B. Pontes, R. Giráldez, J.S. Aguilar-Ruiz, Biclustering on expression data: a review, J. Biomed. Inf. 57 (2015) 163–180.

[6] V. Melnykov, Model-based biclustering of clickstream data, Comput. Stat. Data Anal. 93 (2016) 31–45.

[7] A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay, C.A.C. Coello, Survey of multiobjective evolutionary algorithms for data mining: part ii, IEEE Trans. Evol. Comput. 18 (1) (2014) 20–35.

[8] S. Khan, L. Chen, X. Zhe, H. Yan, Feature selection based on co-clustering for effective facial expression recognition, in: Proc. Int Conf on Machine Learning and Cybernetics (ICMLC2016), IEEE, 2016, pp. 48–53.

[9] M. Denitto, L. Magri, A. Farinelli, A. Fusiello, M. Bicego, Multiple structure recovery via probabilistic biclustering, in: Proc. Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR), 2016, pp. 274–284.

[10] M. Denitto, M. Bicego, A. Farinelli, M. Figueiredo, Spike and slab biclustering, Pattern Recognit. 72 (2017) 186–195.

[11] M. Denitto, S. Melzi, M. Bicego, U. Castellani, A. Farinelli, M.A. Figueiredo, Y. Kleiman, M. Ovsjanikov, Region-based correspondence between 3d shapes via spatially smooth biclustering, in: Proc. of Int. Conf. on Computer Vision, 2017, pp. 4260–4269.

[12] A. Prelic, S. Bleuler, P. Zimmermann, A. Wille, P. Bhlmann, W. Gruissem, L. Hennig, L. Thiele, E. Zitzler, Comparison of biclustering methods: a systematic comparison and evaluation of biclustering methods for gene expression data, Bioinformatics 22 (9) (2006) 1122–1129.

[13] G. Getz, E. Levine, E. Domany, Coupled two-way clustering analysis of gene microarray data, Proc. Natl. Acad. Sci. USA 97 (22) (2000) 12079–12084.

[14] A. Farinelli, M. Denitto, M. Bicego, Biclustering of expression microarray data using affinity propagation, in: Proc. Int. Conf. on Pattern Recognition in Bioinformatics (PRIN2011), 2011, pp. 13–24.

[15] S.R. Bulò, M. Pelillo, Dominant-set clustering: a review, Eur. J. Oper. Res. 262 (2017).

[16] M. Pavan, M. Pelillo, Dominant sets and hierarchical clustering, in: Proc. Int. Conf. on Computer Vision, IEEE, 2003, p. 362.

[17] L. Teng, L. Chan, Discovering biclusters by iteratively sorting with weighted correlation coefficient in gene expression data, J. Signal Process. Syst. 50 (3) (2008) 267–280.

[18] C. Ding, Y. Zhang, T. Li, S.R. Holbrook, Biclustering protein complex interactions with a biclique finding algorithm, in: Proc. Int. Conf on Data Mining (ICDM06), IEEE, 2006, pp. 178–187.

[19] B. Gao, T.-Y. Liu, X. Zheng, Q.S. Cheng, W.Y. Ma, Consistent bipartite graph co–partitioning for star-structured high-order heterogeneous data co-clustering, in: Proc. of ACM Int. Conf. on Knowledge Discovery in Data Mining, ACM, 2005, pp. 41–50.

[20] W. Ahmad, A. Khokhar, cHawk: an efficient biclustering algorithm based on bipartite graph crossing minimization, VLDB Workshop on Data Mining in Bioinformatics, 2007.

[21] L. Dodero, S. Vascon, V. Murino, A. Bifone, A. Gozzi, D. Sona, Automated multi-subject fiber clustering of mouse brain using dominant sets, Front. Neuroinf. 8 (2015) 87.

[22] M. Denitto, M. Bicego, A. Farinelli, M. Pelillo, Dominant set biclustering, in: Proc. Int. Conf. on Energy Minimization Methods in Computer Vision and Pattern Recognition, 2017, pp. 49–61.

[23] M. Pavan, M. Pelillo, Dominant sets and pairwise clustering, IEEE Trans. Pattern Anal. Mach.Intell. 29 (1) (2007) 167–172.

[24] J.W. Weibull, Evolutionary Game Theory, MIT press, 1997.

[25] A. Oghabian, S. Kilpinen, S. Hautaniemi, E. Czeizler, Biclustering methods: biological relevance and application in gene expression analysis, PloS one 9 (3) (2014) e90801.

[26] M. Tepper, G. Sapiro, A biclustering framework for consensus problems, SIAM J. Imaging Sci. 7 (4) (2014) 2488–2525.

[27] E. Zemene, M. Pelillo, Interactive image segmentation using constrained dominant sets, in: Proc. Eur. Conf. on Computer Vision, 2016, pp. 278–294.

[28] A. Ben-Dor, B. Chor, R. Karp, Z. Yakhini, Discovering local structure in gene expression data: the order-preserving submatrix problem, J. Comput. Biol. 10 (3–4) (2003) 373–384.

[29] M. Denitto, A. Farinelli, M.A. Figueiredo, M. Bicego, A biclustering approach based on factor graphs and the max-sum algorithm, Pattern Recognit. 62 (2017) 114–124.

[30] K. Tu, X. Ouyang, D. Han, V. Honavar, Exemplar-based robust coherent biclustering, in: Proc. SIAM Int. Conf. on Data Mining, 2011, pp. 884–895.

[31] A.W. Fitzgibbon, A. Zisserman, Multibody structure and motion: 3-d reconstruction of independently moving objects, in: Proc. European Conf. on Computer Vision (ECCV2000), Springer, 2000, pp. 891–906.

[32] C. Häne, C. Zach, B. Zeisl, M. Pollefeys, A patch prior for dense 3d reconstruction in man-made environments, in: Proc. Int. Conf. on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012, pp. 563–570.

[33] R. Toldo, A. Fusiello, Image-consistent patches from unstructured points with j-linkage, Image Vis. Comput. 31 (10) (2013) 756–770.

[34] M. Soltanolkotabi, E. Elhamifar, E.J. Candès, Robust subspace clustering, Ann. Statist. 42 (2) (2014) 669–699.

[35] L. Magri, A. Fusiello, T-linkage: a continuous relaxation of j-linkage for multi--model fitting, in: Proc. of Int. Conf. on Computer Vision and Pattern Recognition, 2014, pp. 3954–3961.

[36] L. Magri, A. Fusiello, Robust multiple model fitting with preference analysis and low-rank approximation, in: Proc. of British Machine Vision Conference (BMVC), 2015, pp. 20.1–20.12.

[37] T.-T. Pham, T.-J. Chin, J. Yu, D. Suter, The random cluster model for robust geometric fitting, IEEE Trans. Pattern Anal. Mach.Intell. 36 (2014) 1658–1671.

[38] D.S. Cheng, V. Murino, M. Figueiredo, Clustering under prior knowledge with application to image segmentation, in: Advances in Neural Information Processing Systems 19, MIT Press, 2007, pp. 401–408.

[39] V. Ganapathi-Subramanian, B. Thibert, M. Ovsjanikov, L. Guibas, Stable region correspondences between non-isometric shapes, Comput. Graph. Forum 35 (5) (2016) 121–133.

[40] F. Bogo, J. Romero, M. Loper, M.J. Black, FAUST: dataset and evaluation for 3D mesh registration, in: Proc. Int. Conf. on Computer Vision and Pattern Recognition, 2014, pp. 3794–3801.

[41] V.G. Kim, Y. Lipman, T. Funkhouser, Blended intrinsic maps, ACM Trans. Graph. (TOG) 30 (4) (2011) 79.

[42] V. Surazhsky, T. Surazhsky, D. Kirsanov, S.J. Gortler, H. Hoppe, Fast exact and approximate geodesics on meshes, ACM Trans. Graph. (TOG) 24 (3) (2005) 553–560.

[43] R. Henriques, S. Madeira, BiC2PAM: constraint-guided biclustering for biological data analysis with domain knowledge, Algorithms Mol. Biol. 11 (2016) 23.

[44] M. Bicego, P. Lovato, A. Ferrarini, M. Delledonne, Biclustering of expression microarray data with topic models, in: Proc. of Int. Conf. on Pattern Recognition (ICPR2010), 2010, pp. 2728–2731.

[45] A. Perina, P. Lovato, V. Murino, M. Bicego, Biologically-aware latent dirichlet allocation (BaLDA) for the classification of expression microarray, in: Proc. Int. Conf. on Pattern Recognition in Bioinformatics (PRIB2010), 2010, pp. 230–241.

Matteo Denitto received his bachelor and master degree in Bioinformatics from the University of Verona in 2011 and 2013 respectively. He obtained his PhD in Computer Science at the University of Verona in 2017. His research interests involve biclustering, graphical models and factor graphs.

Manuele Bicego received his Laurea degree and PhD degree in Computer Science from University of Verona in 1999 and 2003, respectively. From 2004 to 2008 he was at the University of Sassari, in the Computer Vision Lab. Currently he is assistant professor (ricercatore) at the University of Verona, and member of the VIPS (Vision Image Processing & Sound) lab at the Computer Science Department. From June 2009 to February 2011 he was also member of the PLUS (Pattern analysis, Learning and image Understanding Systems) lab at the Istituto Italiano di Tecnologia (IIT - Genova Italy). His research interests include statistical pattern recognition, mainly probabilistic models (GMM, HMM) and kernel machines (e.g. SVM), with application to video analysis, biometrics and, recently, bioinformatics. Manuele Bicego is author of several papers in the above subjects, published in international journals and conferences. He is an associate editor of ELCVIA (Jan 2014 -) and Pattern Recognition (Jul 2016 -). He has served as member of the scientific committee of different international conferences, and he is a reviewer for several international conferences and journals. Manuele Bicego is member of the IEEE Systems, Man, and Cybernetics society and of the IAPR Society Italian Chapter (GIRPR).

Alessandro Farinelli is associate professor at University of Verona, Department of Computer Science, since December 2014. His research interests comprise theoretical and practical issues related to the development of Artificial Intelligent Systems applied to robotics. In particular, he focuses on coordination, decentralised optimisation and information integration for Multi-Agent and Multi-Robot systems, control and evaluation of autonomous mobile robots. He was principal investigator for several national and international research projects in the broad area of Artificial Intelligence for robotic systems. He co-authored more than 80 peer-reviewed scientific contributions in top international journals (such AIJ) and conferences (such as IJCAI, AAMAS, and AAAI).

Sebastiano Vascon is currently a PhD student at the Pattern Analysis and Computer Vision department at the Istituto Italiano di Tecnologia of Genova. He studied Computer Science at the University Ca Foscari of Venice where he received the BSc degree in 2009 and the MSc degree cum laude in 2012 under the supervision of prof. Pelillo and prof. Torsello. During the MSc he spent 6 months at the University College of London under the Erasmus project. His main interests are on pattern recognition, computer vision, graph-theory and game-theory with applications in clustering, medical imaging, scene understanding and behavior analysis.

Marcello Pelillo is a Professor of Computer Science at the University of Venice, Italy, where he directs the European Centre for Living Technology and leads the Computer Vision and Pattern Recognition group, which he founded in 1995. He held visiting research positions at Yale University (USA), McGill University (Canada), the University of Vienna (Austria), York University (UK), the University College London (UK), and the National ICT Australia (NICTA) (Australia). He serves (or has served) on the editorial boards of IEEE Transactions on Pattern Analysis and Machine Intelligence, IET Computer Vision, Pattern Recognition, Brain Informatics, and is on the advisory board of the International Journal of Machine Learning and Cybernetics. He has initiated several conferences series as Program Chair (EMMCVPR, IWCV, SIMBAD) and will serve as a General Chair for ICCV 2017. He is (or has been) scientific coordinator of several research projects, including SIMBAD, a highly successful EU-FP7 project devoted to similarity-based pattern analysis and recognition. Prof. Pelillo has been elected a Fellow of the IEEE and a Fellow of the IAPR, and has been appointed IEEE Distinguished Lecturer (2016–2017 term). His Erdos number is 2.