Machine Learning
https://doi.org/10.1007/s10994-024-06607-x

Self-organizing maps with adaptive distances for multiple dissimilarity matrices

Laura Maria Palomino Mariño[1] · Francisco de Assis Tenorio de Carvalho[1]

Received: 27 December 2022 / Revised: 10 July 2024 / Accepted: 4 August 2024

Abstract
There has been an increasing interest in multi-view approaches based on their ability to manage data from several sources. However, regarding unsupervised learning, most multi-view approaches are clustering algorithms suitable for analyzing vector data. Currently, only a relatively few SOM algorithms can manage multi-view dissimilarity data, despite their usefulness. This paper proposes two new families of batch SOM algorithms for multi-view dissimilarity data: multi-medoids SOM and relational SOM, both designed to give a crisp partition and learn the relevance weight for each dissimilarity matrix by optimizing an objective function, aiming to preserve the topological properties of the map data. In both families, the weight represents the relevance of each dissimilarity matrix for the learning task being computed, either locally, for each cluster, or globally, for the whole partition. The proposed algorithms were compared with already in the literature single-view SOM and set-medoids SOM for multi-view dissimilarity data. According to the experiments using 14 datasets for F-measure, NMI, Topographic Error, and Silhouette, the relevance weights of the dissimilarity matrices must be considered. In addition, the multi-medoids and relational SOM performed better than the set-medoids SOM. An application study was also carried out on a dermatology dataset, where the proposed methods have the best performance.

Keywords Self-organizing maps · Batch SOM · Multi-view dissimilarity data · Relevance weights · Adaptive distances

Editors: Michelangelo Ceci, João Gama, Jose Lozano, André de Carvalho, Paula Brito.

Laura Maria Palomino Mariño and Francisco de Assis Tenorio de Carvalho have contributed equally to this work.

✉ Francisco de Assis Tenorio de Carvalho
fatc@cin.ufpe.br

Laura Maria Palomino Mariño
lmpm@cin.ufpe.br

[1] Centro de Informatica, Universidade Federal de Pernambuco, Av. Jornalista Anibal Fernandes s/n - Cidade Universitaria, Recife, PE 50740-560, Brazil

1 Introduction

Multi-view data research has become increasingly important since large amounts of information are constantly being generated from different sources. These data are heterogeneous, and the variables are organically partitioned into groups; however, there are still some potential links between them. Each group of variables is referred to as a specific perspective (view), and the multiple views on a given issue can take different arrangements (Xu et al., 2015). Often, in multi-view data, the objects are represented by several matrices or views (e.g., sensor signal data can be decomposed into time and frequency data, image data can be described by texture and color data, and multimedia segments can be represented by video and audio signal data (Yang & Wang, 2018)). In such data, each view has its specific features for a certain knowledge discovery task. However, different views frequently contain complementary information that must be exploited (Wang et al., 2017).

According to Sun et al. (2019), multi-view learning can improve learning performance effectively on natural single-view data (Sun et al., 2019), thus representing the main reason for single-view-based data representation to be usually incomplete. In this sense, different views might provide complementary information for the learning problem (Wang et al., 2017). Concerning multi-view clustering, there are three combination strategies to perform the learning task (Cleuziou et al., 2009): concatenation strategy, distributed strategy, and centralized strategy. The first strategy consists of concatenating the views into a single one, either directly, by juxtaposing the sets of features, or indirectly, by combining the proximity matrices derived from each view. The second strategy starts by clustering the objects from each view independently and then looking for a solution that represents a consensus among all the groups. Finally, the last strategy uses multiple views simultaneously to mine hidden patterns from the data (Cleuziou et al., 2009).

Throughout learning, the multi-view approaches explicitly use distinct data representations that can either be the original data features or those obtained through computations (Sun et al., 2019). Moreover, each view can be represented by either vectorial or non-vectorial data. The former has received considerably more attention from unsupervised learning approaches, where most machine learning and data analysis methods available are based on a vector model, with each example being represented by a vector of quantitative values (Frigui et al., 2007). Unfortunately, many current datasets do not support this type of representation. There are categorical data, abstract data, and relational data, among others. In the latter, the objects are described through a relationship between data pairs that contain only information on the degrees to which pairs of objects in the dataset are related (Kaufman & Rousseeuw, 1987; Frigui et al., 2007). A way of dealing with these types of data is to consider the objects represented by a matrix of dissimilarities.

In dissimilarity data, each pair of objects is represented by a dissimilarity relationship (Rousseeuw & Kaufman, 2005). Single-view dissimilarity data is represented by a dissimilarity matrix defined as $\mathbf{D} = [d(e_k, e_l)]\,(1 \leq k, l \leq N)$, where $d(e_k, e_l)$ is the dissimilarity between objects e_k and e_l on dissimilarity matrix $\mathbf{D}$. Unsupervised dissimilarity data methods introduce approaches to handle such data using the most appropriate dissimilarity function for the problem at hand. These approaches can handle heterogeneous data through different transformations.

Therefore, dissimilarity data are more generic since they can be applied to scenarios where objects cannot be represented by numerical features. They are also more useful when the distance measure is computed according to a suitable algorithm instead of an algebraic expression as usual, or when sets of similar objects cannot be represented adequately by a prototype vector (Frigui et al., 2007).

Several traditional machine learning algorithms have been extended to cope effectively with a variety of multi-view and dissimilarity data problems (Gusmão & Carvalho, 2019; Dantas & Carvalho, 2011; Olteanu & Villa-Vialaneix, 2015). The Kohonen Self-Organizing Map (SOM) (Kohonen, 2001; Badran et al., 2005; Kohonen, 2013; Astudillo & Oommen, 2014; Cottrell et al., 2018) is a powerful tool for dealing with these kinds of challenges. SOM performs clustering and non-linear data projection at the same time, thus providing a strong visualization tool. The SOM has proven effective, especially when considering the faithfulness (precision) of the mapping from a high-dimensional space (Vatanen et al., 2015). In addition, the SOM has proven a reliable approach for a variety of fields, including finance, statistics, bio-medicine, industry, and many others (see Refs. Kaski et al. (1998); Oja et al. (2003); Domínguez-González et al. (2012); Astudillo and Oommen (2014); Kamimura (2019); Douzas et al. (2021) for details).

A SOM consists of neurons (vertices) usually arranged on a regular two or three-dimensional grid (the map). Each neuron is associated with a cluster representative (prototype) of a data subset (a cluster). Both the data and the *a priori* topology impose the cluster structure. The SOM network can be trained either incrementally or batch-wise. According to Kohonen (2013), the batch variant of the SOM network is most suited for practical applications. However, incremental training is the preferable approach when data are given sequentially. Throughout the map training, each object must select its Best Matching Unity (BMU), i.e., the neuron with the most similar prototype to its description. Therefore, the BMU-associated prototype and the neuron-associated prototypes in the BMU spatial neighborhood are updated to better represent the object similarity to these neurons.

A SOM preserves the data topological properties, which implies that if two objects in the original description space are similar, the related BMU prototypes are also similar and will be associated with adjacent or close vertices on the map. Thus, the data are grouped by clusters such that the most similar prototypes are associated with adjacent vertices, while less similar prototypes are associated with distant vertices on the map (Kohonen, 2013).

There are different prototype-based approaches to determine the representative of a cluster concerning dissimilarity data, but only a few SOM algorithms have been proposed. Median Batch SOM (Kohonen, 2001) was the first extension of the original SOM for single-view dissimilarity data. In this regard, the cluster prototypes were represented by a single medoid in Ref. Kohonen (2001) and later by a set-medoids in Ref. Golli et al. (2005). Furthermore, another method for extending SOM to single-view dissimilarity data was proposed, in which the cluster representative is a "normalized linear combination" of the objects from the whole dataset. Based on this latter approach, batch (Hasenfuss & Hammer, 2007) and on-line (Olteanu et al., 2012) versions of SOM for dissimilarity data are currently available. Finally, Ref. Mariño and Carvalho (2020) introduced a batch SOM for single-view dissimilarity data, with the cluster prototypes represented by vectors of weights. Each component of these vectors quantifies the significance of each object in a given cluster. Despite the growing interest in machine learning, to the best of our knowledge, only Refs. Dantas and Carvalho (2011) and Olteanu and Villa-Vialaneix (2015) have proposed SOM algorithms that can manage multi-view dissimilarity data. Ref. Olteanu and Villa-Vialaneix (2015) introduced an online extension of relational SOM algorithm (Hasenfuss & Hammer, 2007), whereas Refs. Dantas and Carvalho (2011) proposed a batch extension of the median SOM (Golli et al., 2005; Kohonen, 2001) to the case where several dissimilarities matrices are available for describing the dataset.

Paper proposal

Herein, we propose two families of batch SOM algorithms for multi-view dissimilarity data in the framework of the centralized strategy: Batch SOM algorithms for multi-view dissimilarity data with weighted medoids as cluster representatives (MBSOM-MM**dd**) and Batch SOM algorithms for multi-view dissimilarity data with a normalized linear combination of the objects as cluster representatives (MRBSOM).

The new MBSOM-MM**dd** family extends Ref. Mariño and Carvalho (2020) aiming to manage datasets described by multiple dissimilarity matrices. For a fixed neighborhood, the new method gives the optimal solution for computing the representative (weighted medoids) associated with each neuron, computing optimal adaptive relevance weights on the dissimilarity matrices, and providing the optimal cluster partition.

Furthermore, MRBSOM, as the already in the literature batch SOM methods for relational data (Hasenfuss & Hammer, 2007), keeps the idea that each cluster representative is a normalized linear combination of the objects represented in the description space, but additionally uses different adaptive weights on the dissimilarity matrices aiming to take into account the importance of each dissimilarity matrix on the unsupervised learning task.

In the families of both models, the weights change at each algorithm iteration such that each matrix has a different influence on the training of the map. Each one of the proposed families can compute relevance weights for each dissimilarity matrix, either locally, for each cluster, or globally, for the whole partition (Dantas & Carvalho, 2011).

The main contributions of our paper are the two families of Batch SOM algorithms that can manage multi-view datasets described by several dissimilarity matrices. More precisely, the paper provides:

- The respective objective functions that, for a fixed neighborhood, should be optimized to learn the MBSOM-MM**dd** and MRBSOM model families;
- For a fixed neighborhood radius, i) the optimal solution for computing the cluster representatives associated with each neuron in the proposed models; ii) the optimal solution for computing the relevance weights of the dissimilarity matrices on the training of the SOM; and iii) the optimal solution for the partition associated to the neurons of the proposed algorithms;
- The time complexity of the proposed models;
- A significant evaluation of the proposed methods compared with relevant batch SOM algorithms for multi-view dissimilarity data.

Thus, the proposed algorithms aim to improve MBSOM-CM**dd** (Dantas & Carvalho, 2011) because the number of objects (medoids) that represent a cluster may be insufficient to describe it. Moreover, MBSOM-CM**dd** ignores the relevance of the medoids, for instance, when several objects are selected as medoids, these medoids are not necessarily equally important for the cluster. They do not describe it in the same way. Additionally, in MBSOM-CM**dd**, the cardinality of the set of medoids (the representative) is a parameter that must be provided *a priori*. Finally, the relevance of the specific data source can impact the result of model performance. Nevertheless, the impact of data sources considering the relevance for the cluster locally and for each view globally has not been studied with these approaches of cluster representation regarding SOM algorithms.

The paper is organized as follows: Sect. 2 describes the families of the proposed models MBSOM-CM**dd** and MRBSOM. In addition, we also provide an in-depth description and formalization of the algorithm MBSOM-CM**dd** introduced in the work of Ref. Dantas and Carvalho (2011). Moreover, we analyze the time complexity of the proposed algorithms.

Section 3 presents the setup of our experiments. Section 4 provides the performance evaluation of the proposed algorithms against already in the literature approaches, showing the results and discussing the main findings obtained. This section also provides further insights into the families of the models through an application concerning the dermatology dataset (Dua & Graff, 2017). Finally, Sect. 5 introduces our final remarks.

2 Batch SOM algorithms for multi-view dissimilarity data

This section presents the batch SOM families for multi-view dissimilarity data MBSOM-CM**dd**, MBSOM-MM**dd**, and MRBSOM. Section 2.1 discusses the cluster representatives and the relevance weights of the dissimilarity matrices and provides the error functions of these batch SOM algorithms. Section 2.2 describes their three main steps (the computation of the cluster representatives, the relevance weights of the dissimilarity matrices, and the update of the clusters) and provides the main algorithm for each batch SOM family. Section 2.3 introduces some notations aiming to simplify the presentation of the methods according to the algorithms used and the relevance weights assigned to each dissimilarity matrix. Finally, Sect. 2.4 introduces the time complexity analysis of the proposed variants of the batch SOM families.

2.1 MBSOM-CMdd, MBSOM-MMdd and MRBSOM self-organizing maps

This section provides a detailed presentation of the MBSOM-CM**dd**, MBSOM-MM**dd**, and MRBSOM SOM algorithms.

Let $E = \{e_1, \dots, e_N\}$ be a set of objects and let P dissimilarity matrices $\mathbf{D}_p = [d_p(e_k, e_l)]$ $(1 \leq k, l \leq N)$ where $d_p(e_k, e_l)$ is the dissimilarity between objects e_k and e_l on dissimilarity matrix $\mathbf{D}_p$ $(1 \leq p \leq P)$.

A SOM consists of a low-dimensional (usually two-dimensional) regular grid (map), which contains C nodes (neurons). Each SOM map is associated with a partition in which a neuron indexed by r has associated a cluster P_r and a representative (prototype).

Let $\{1, \dots, C\}$ be the cluster index set and let f be the assignment function that maps each object to an index $r = f(e_k) \in \{1, \dots, C\}$ of the cluster index set. The partition $\mathcal{P} = \{P_1, \dots, P_C\}$ associated with a SOM is defined by the assignment function which gives the index of the cluster of $\mathcal{P}$ to which the object e_k belongs to, i.e., $P_r = \{e_k \in E : f(e_k) = r\}$.

Following (Golli et al., 2005), in MBSOM-CM**dd** (Dantas & Carvalho, 2011) it is assumed that the representative of each cluster is a set of objects (set-medoids), i.e., the prototype G_r of cluster P_r is a subset of fixed cardinal $1 \leq q \ll N$ of the set of objects E: $G_r \in E^{(q)} = \{A \subset E : |A| = q\}$. Besides, $\mathcal{G} = (G_1, \dots, G_r, \dots, G_C)$ is the vector of cluster prototypes.

Moreover, following (Mariño & Carvalho, 2020), in MBSOM-MM**dd** it is assumed that the representative $\mathbf{v}_r = (v_{r1}, \dots, v_{rN})$ of cluster P_r is a N-dimensional vector of weights whose components measure how the objects are weighted as a medoid regarding the cluster P_r. Let $\mathbf{V} = (\mathbf{v}_1, \dots, \mathbf{v}_C) = (v_{rj})$ $(1 \leq r \leq C; 1 \leq j \leq N)$ be the matrix of prototype weights of the objects regarding the clusters.

Regarding MRBSOM, Refs. Cottrell et al. (2018); Hammer and Hasenfuss (2007) pointed out that if the data are described by a dissimilarity matrix where each cell is the squared Euclidean distance, they can be embedded in a pseudo-Euclidean space in such a way that optimum prototypes can be expressed as linear combinations of data points. Therefore, the unknown distances $||\mathbf{x}_k - \mathbf{v}_r||^2$, where $\mathbf{x}_k = (\mathbf{x}_{k(1)}, \dots, \mathbf{x}_{k(P)})$ is the description of the object e_k and $\mathbf{v}_r = (\mathbf{v}_{r(1)}, \dots, \mathbf{v}_{r(P)})$ is the representative of cluster P_r both in the pseudo-Euclidean space, can be expressed in terms of known values of the squared Euclidean distances of the dissimilarity matrix.

Assuming that $||\mathbf{x}_k - \mathbf{v}_r||^2 = \sum_{p=1}^{P} ||\mathbf{x}_{k(p)} - \mathbf{v}_{r(p)}||^2$ and that the p-th component of the cluster representative is such that $\mathbf{v}_{r(p)} = \sum_{k=1}^{N} \alpha_{rk}\mathbf{x}_{k(p)}$ where $\sum_{k=1}^{N} \alpha_{rk} = 1$, according to Cottrell et al. (2018); Hasenfuss and Hammer (2007):

$$||\mathbf{x}_{k(p)} - \mathbf{v}_{r(p)}||^2 = [\mathbf{D}_p\boldsymbol{\alpha}_r]_k - \frac{1}{2}\boldsymbol{\alpha}_r^\top \mathbf{D}_p \boldsymbol{\alpha}_r \, (1 \leq p \leq P), \tag{1}$$

where $[\mathbf{D}_p\boldsymbol{\alpha}_r]_k$ is the k-th component of $[\mathbf{D}_p\boldsymbol{\alpha}_r]$ and $\boldsymbol{\alpha}_r = (\alpha_{r1}, \dots, \alpha_{rN})\,(1 \leq r \leq C)$. Since $\mathbf{v}_{r(p)}$ is in the implicitly pseudo-Euclidean space, it is the vector $\boldsymbol{\alpha}_r$ that is updated, where the distances between the prototypes and the objects are computed only indirectly through the coefficients α_{rk}. According to Ref. Hammer and Hasenfuss (2007), the equation (1) still holds to any given dissimilarity matrix $\mathbf{D}_p$.

Therefore, in MRBSOM it is assumed that the representative $\boldsymbol{\alpha}_r = (\alpha_{r1}, \dots, \alpha_{rN})$ of cluster P_r is a N-dimensional vector of coefficients α_{rk}. Let $\mathcal{A} = (\boldsymbol{\alpha}_1, \dots, \boldsymbol{\alpha}_C) = (\alpha_{rk})_{\substack{1 \leq r \leq C \\ 1 \leq k \leq N}}$ be the matrix of coefficients α_{rk}.

Dissimilarity matrices can have different relevance to the training of the SOM. In most applications, some dissimilarity matrices may be irrelevant, while among those that are relevant, some may be more or less relevant than others.

Therefore, aiming to obtain a significant SOM from all dissimilarity matrices, the MBSOM-CM**dd**, MBSOM-MM**dd**, and MRBSOM SOM algorithms were designed to provide the clusters and their respective prototypes by simultaneously preserving the spatial order of the prototypes on the map, as well as to learn the relevance weight for each dissimilarity matrix by optimizing a suitable error function.

The relevance weights can be assigned to each dissimilarity matrix globally, to all the clusters, according to the $(P \times 1)$ matrix $\mathbf{W} = (w_p)\,(1 \leq p \leq P)$, with $w_p \in \mathrm{R}^+$. They can also be assigned to each dissimilarity locally, to each cluster, according to the $(P \times C)$ matrix $\mathbf{W} = (w_{rp})\,(1 \leq p \leq P; 1 \leq r \leq C)$, with $w_{rp} \in \mathrm{R}^+$.

The training of the MBSOM-CM**dd** algorithm provides the vector of prototypes $\mathcal{G}$, the matrix of relevance weights $\mathbf{W}$, and the partition $\mathcal{P}$ by iteratively minimizing the error function $J_{MBSOM-MMdd}$. Furthermore, the training of the MBSOM-MM**dd** algorithm provides the matrix $\mathbf{V}$ of prototype weights, the matrix of relevance weights $\mathbf{W}$, and the partition $\mathcal{P}$ by iteratively minimizing the error function $J_{MBSOM-MMdd}$. Furthermore, the training of the MRBSOM algorithm provides the matrix $\mathcal{A}$ of coefficients, the matrix of relevance

Table 1 Error functions of the SOM algorithms

Algorithms	Error functions
MBSOM-CM**dd** (Dantas & Carvalho, 2011)	$J_{MBSOM-MMdd}(\mathcal{G}, \mathbf{W}, \mathcal{P}) = \sum_{k=1}^{N} \Delta_{\mathbf{W}}(e_k, G_{f(e_k)}).$ (2)
MBSOM-MM**dd**	$J_{MBSOM-MMdd}(\mathbf{V}, \mathbf{W}, \mathcal{P}) = \sum_{k=1}^{N} \Delta_{\mathbf{W}}(e_k, \mathbf{v}_{f(e_k)}).$ (3)
MRBSOM	$J_{MRBSOM}(\mathcal{A}, \mathbf{W}, \mathcal{P}) = \sum_{k=1}^{N} \Delta_{\mathbf{W}}(e_k, \boldsymbol{\alpha}_{f(e_k)}).$ (4)

Table 2 Generalized dissimilarity functions of the SOM algorithms

Algorithms	Generalized dissimilarity functions
MBSOM-CM**dd** (Dantas & Carvalho, 2011)	$\Delta_{\mathbf{W}}(e_k, G_{f(e_k)}) = \sum_{r=1}^{C} h_{f(e_k),r} D_{\mathbf{W}}(e_k, G_r).$ (5)
MBSOM-MM**dd**	$\Delta_{\mathbf{W}}(e_k, \mathbf{v}_{f(e_k)}) = \sum_{r=1}^{C} h_{f(e_k),r} D_{\mathbf{W}}(e_k, \mathbf{v}_r).$ (6)
MRBSOM	$\Delta_{\mathbf{W}}(e_k, \boldsymbol{\alpha}_{f(e_k)}) = \sum_{r=1}^{C} h_{f(e_k),r} D_{\mathbf{W}}(e_k, \boldsymbol{\alpha}_r).$ (7)

Table 3 Global match functions: the weights are assigned globally

Algorithms	Global match functions
MBSOM-CM**dd** (Dantas & Carvalho, 2011)	$D_{\mathbf{W}}(e_k, G_r) = \sum_{p=1}^{P} w_p D_p(e_k, G_r).$ (9)
MBSOM-MM**dd**	$D_{\mathbf{W}}(e_k, \mathbf{v}_r) = \sum_{p=1}^{P} w_p D_p(e_k, \mathbf{v}_r).$ (10)
MRBSOM	$D_{\mathbf{W}}(e_k, \boldsymbol{\alpha}_r) = \sum_{p=1}^{P} w_p D_p(e_k, \boldsymbol{\alpha}_r).$ (11)

weights $\mathbf{W}$, and the partition $\mathcal{P}$ by iteratively minimizing the error function J_{MRBSOM}. Table 1 provides the error functions of the algorithms.

For each object e_k, the winning neuron, known as the best matching unit (BMU), is the neuron with the cluster representative closest to e_k. The BMU is indexed by $f(e_k)$ and is identified with the corresponding cluster representative.

The error measure of a BMU regarding the object e_k is computed by the generalized dissimilarity (Badran et al., 2005) function $\Delta_{\mathbf{W}}$, which allows comparing each object e_k to each cluster representative. Table 2 provides $\Delta_{\mathbf{W}}$ according to the SOM algorithms.

In the equations (5), (6) and (7), $h_{f(e_k),r}$ is the neighborhood kernel function that measures the influence neighborhood of BMU on neuron r. Several choices are possible, in Ref Dantas and Carvalho (2011), it is defined as

$$h_{f(\mathbf{e}_k),r} = \exp\left\{ -\frac{\|a_{f(\mathbf{e}_k)} - a_r\|^2}{2\sigma^2} \right\}. \tag{8}$$

where $a_{f(\mathbf{e}_k)}$ and a_r are the BMU and neuron r positions in the grid, respectively. Moreover, σ is the neighborhood radius. The size of the neighborhood decreases with σ: the smaller σ the fewer the neurons belonging to the effective neighborhood of a given BMU (Badran et al., 2005; Kohonen, 2001).

Therefore, the generalized dissimilarity function $\Delta_{\mathbf{W}}$ is a weighted sum of the global match functions $D_{\mathbf{W}}$ computed between an object e_k and a cluster representative. Note that it considers all the neurons in the neighborhood of the BMU.

The function $D_{\mathbf{W}}$ computes the global matching between an object e_k and a cluster representative. Table 3 provides $D_{\mathbf{W}}$ according to the SOM algorithms when the weights are assigned globally.

Table 4 provides $D_{\mathbf{W}}$ according to the SOM algorithms when the weights are assigned locally.

The function D_p computes the local matching between an object e_k and a cluster representative on dissimilarity $\mathbf{D}_p$ $(1 \leq p \leq P)$. Table 5 provides D_p according to the SOM algorithms.

Table 4 Global match functions: the weights are assigned locally

Algorithms	Global match functions
MBSOM-CM**dd** (Dantas & Carvalho, 2011)	$D_{\mathbf{W}}(e_k, G_r) = \sum_{p=1}^{P} w_{rp} D_p(e_k, G_r)$. (12)
MBSOM-MM**dd**	$D_{\mathbf{W}}(e_k, \mathbf{v}_r) = \sum_{p=1}^{P} w_{rp} D_p(e_k, \mathbf{v}_r)$. (13)
MRBSOM	$D_{\mathbf{W}}(e_k, \boldsymbol{\alpha}_r) = \sum_{p=1}^{P} w_{rp} D_p(e_k, \boldsymbol{\alpha}_r)$. (14)

Table 5 Local match functions

Algorithms	Match functions
MBSOM-CM**dd** (Dantas & Carvalho, 2011)	$D_p(e_k, G_r) = \sum_{e \in G_r} d_p(e_k, e) \, (1 \leq p \leq P)$. (15)
MBSOM-MM**dd**	$D_p(e_k, \mathbf{v}_r) = \sum_{j=1}^{N} (v_{rj})^n d_p(e_k, e_j) \, (1 \leq p \leq P)$. (16)
MRBSOM	$D_p(e_k, \boldsymbol{\alpha}_r) = \lVert \mathbf{x}_{k(p)} - \mathbf{v}_{r(p)} \rVert^2 = [\mathbf{D}_p \boldsymbol{\alpha}_r]_k - \frac{1}{2} \boldsymbol{\alpha}_r^{\top} \mathbf{D}_p \boldsymbol{\alpha}_r \, (1 \leq p \leq P)$. (17)

Note that because the global match functions are weighted sums of the local match functions, the different dissimilarity matrices need to have been scaled previously by a suitable normalization to make their relative value ranges comparable, thus preventing some particular views from prevailing above the others in the computation of the global match functions only due to different feature measurement units.

2.2 MBSOM-CMdd, MBSOM-MMdd and MRBSOM algorithms

For a fixed radius σ, from an initial solution, the training map of the SOM algorithms are obtained by minimizing the error functions of Table 1, which are performed iteratively in three steps: representation, weighting, and assignment.

The representation step gives the optimal solution for computing the cluster representatives associated with the map neurons. The weighting step computes the relevance weights of the dissimilarity matrices to the training of the SOM. Finally, the assignment step provides the optimal solution for the clusters associated with the map neurons.

2.2.1 Representation step

The representation step gives the optimal solution for computing the cluster representatives associated with the map neurons. During the representation step, the matrix of relevance weights $\mathbf{W}$ and the partition $\mathcal{P}$ are kept fixed.

MBSOM-CM* dd *algorithm The error function $J_{MBSOM-MMdd}$ is minimized regarding the vector of prototypes $\mathcal{G} = (G_1, \dots, G_C)$.

The prototype G_r of cluster P_r, which minimizes the error function $J_{MMBSOM-MMdd}$, either minimizes $\sum_{k=1}^{N} h_{f(e_k),r} \sum_{p=1}^{P} w_p D_p(e_k, G_r)$ (if the weights are assigned globally) or minimizes $\sum_{k=1}^{N} h_{f(e_k),r} \sum_{p=1}^{P} w_{rp} D_p(e_k, G_r)$ (if the weights are assigned locally). The prototype G_r is computed according to the following brute force algorithm 1 (Dantas & Carvalho, 2011):

Table 6 Lagrangian functions

Weights	Lagrangian functions
Global	$\mathcal{L} = \sum_{k=1}^{N} \sum_{r=1}^{C} h_{f(e_k),r} \sum_{p=1}^{P} w_p \sum_{j=1}^{N} (v_{rj})^n d_p(e_k, e_j) - \sum_{r=1}^{C} \beta_r \Big(\sum_{j=1}^{N} v_{rj} - 1 \Big).$ (18)
Local	$\mathcal{L} = \sum_{k=1}^{N} \sum_{r=1}^{C} h_{f(e_k),r} \sum_{p=1}^{P} w_{rp} \sum_{j=1}^{N} (v_{rj})^n d_p(e_k, e_j) - \sum_{r=1}^{C} \beta_r \Big(\sum_{j=1}^{N} v_{rj} - 1 \Big).$ (19)

Algorithm 1

```
1: for $r = 1$ to $C$ do
2:     for $l = 1$ to $N$ do
3:         Global weights: compute $g(e_l) = \sum_{k=1}^{N} h_{f(e_k),r} \sum_{p=1}^{P} w_p \, d_p(e_k, e_l)$
4:         Local weights: compute $g(e_l) = \sum_{k=1}^{N} h_{f(e_k),r} \sum_{p=1}^{P} w_{rp} \, d_p(e_k, e_l)$
5:     end for
6:     Let $e_{(1)}, \ldots, e_{(N)} : g(e_{(1)}) \leq \ldots \leq g(e_{(N)})$
7:     Set $G_r \leftarrow \{e_{(l)}, \ldots, e_{(q)}\}$
8: end for
```

MBSOM-MM* dd *algorithm

The error function $J_{MBSOM-MMdd}$ is minimized regarding the matrix $\mathbf{V} = (v_{rj})\,(1 \leq r \leq C; 1 \leq j \leq N)$ of prototype weights of the objects. To exclude the trivial solution $\mathbf{V}$ equal to null, we consider the sum constraint. Therefore, the error function $J_{MBSOM-MMdd}$ is minimized regarding the matrix $\mathbf{V} = (v_{rj})\,(1 \leq r \leq C; 1 \leq j \leq N)$ of prototype weights of the objects, subject to $\sum_{j=1}^{N} v_{rj} = 1$ and $v_{rj} \geq 0$, either if the weights are assigned globally or locally.

Table 6 provides the Lagrangian functions either if the weights are assigned globally or locally, where β_r are the Lagrange multipliers.

Then, taking the partial derivatives of $\mathcal{L}$ w.r.t v_{rj} and β_r, and by setting the partial derivatives to zero, we obtain the optimal solutions to v_{rj}, which are shown in Table 7.

Remark Equation (20) allows us to conclude that at the end of the training of MBSOM-CM**dd** algorithm when $h_{f(e_k),r} \sim 0$ for $f(e_k) \neq r$, the lower the $\sum_{e_k \in P_r} \sum_{p=1}^{P} w_{rp} d_p(e_k, e_j)$ is, the higher the prototype weight v_{rj} is. In addition, Eq. (21) demonstrates that the lower the $\sum_{e_k \in P_r} \sum_{p=1}^{P} w_p d_p(e_k, e_j)$ is, the higher the prototype weight v_{rj} is.

MRBSOM algorithm

Table 7 Prototype weights of the objects

Weights	Prototype weights
Global	$v_{rj} = \left[\sum_{s=1}^{N} \left(\frac{\sum_{k=1}^{N} h_{f(e_k),r} \sum_{p=1}^{P} w_p d_p(e_k,e_j)}{\sum_{k=1}^{N} h_{f(e_k),r} \sum_{p=1}^{P} w_p d_p(e_k,e_s)} \right)^{\frac{1}{n-1}} \right]^{-1}.$ (20)
Local	$v_{rj} = \left[\sum_{s=1}^{N} \left(\frac{\sum_{k=1}^{N} h_{f(e_k),r} \sum_{p=1}^{P} w_{rp} d_p(e_k,e_j)}{\sum_{k=1}^{N} h_{f(e_k),r} \sum_{p=1}^{P} w_{rp} d_p(e_k,e_s)} \right)^{\frac{1}{n-1}} \right]^{-1}.$ (21)

The error function J_{MRBSOM} is minimized regarding the matrix $\mathcal{A} = (\alpha_{rk})\,(1 \leq r \leq C; 1 \leq k \leq N)$ of coefficients. The error function J_{MRBSOM} can be written as

$$J_{MRBSOM} = \sum_{k=1}^{N} \sum_{r=1}^{C} h_{f(e_k),r} \sum_{p=1}^{P} (\mathbf{x}_{k(p)} - \mathbf{v}_{r(p)})^{\top} (\mathbf{x}_{k(p)} - \mathbf{v}_{r(p)}). \tag{22}$$

Taking the partial derivative of J_{MRBSOM} w.r.t $\mathbf{v}_{r(p)}$ and by setting the partial derivative to zero, we obtain:

$$\mathbf{v}_{r(p)} = \sum_{k=1}^{N} \frac{h_{f(e_k),r}}{\sum_{k=1}^{N} h_{f(e_k),r}} \mathbf{x}_{k(p)} = \sum_{k=1}^{N} \alpha_{rk} \mathbf{x}_{k(p)}. \tag{23}$$

where $\alpha_{rk} = \frac{h_{f(e_k),r}}{\sum_{k=1}^{N} h_{f(e_k),r}}$, considering that $\sum_{k=1}^{N} \alpha_{rk} = 1$, and $\alpha_{rk} \geq 0\,(1 \leq r \leq C; 1 \leq k \leq N)$. Since $\mathbf{v}_{r(p)}$ is in the pseudo-Euclidean space, it is the matrix $\mathcal{A} = (\alpha_{rk})\,(1 \leq r \leq C; 1 \leq k \leq N)$ of coefficients that is updated as follows:

$$\alpha_{rk} = \frac{h_{f(e_k),r}}{\sum_{k=1}^{N} h_{f(e_k),r}}\,(1 \leq r \leq C; 1 \leq k \leq N). \tag{24}$$

2.2.2 Weighting step

During the weighting step, the cluster representatives and the partition $\mathcal{P}$ are kept fixed. The error functions of Table 1 are minimized regarding the matrix of weights $\mathbf{W}$.

A trivial solution for this minimization problem is reached when $\mathbf{W}$ is null. To exclude the trivial solution, constraints on the elements of $\mathbf{W}$ are required. Two main types of constraints have been proposed: a constraint on the product of the weights (Diday & Govaert, 1977) and a constraint on the sum of the weights (Huang et al., 2005). Herein, we consider only the product constraint since the sum constraint requires fixing further hyper-parameters in advance.

Therefore, the error functions of Table 1 are minimized either regarding the matrix of weights $\mathbf{W} = (w_p)\,(1 \leq p \leq P)$, subject to $\prod_{p=1}^{P} w_p = 1$, $w_p > 0$, if the weights are assigned globally, or regarding the matrix of weights $\mathbf{W} = (w_{rp})\,(1 \leq r \leq C; 1 \leq p \leq P)$, subject to $\prod_{p=1}^{P} w_{rp} = 1$, $w_{rp} > 0$, if the weights are assigned locally.

Table 8 provides the Lagrangian functions according to the SOM algorithms for weights assigned globally, where β is the Lagrange multiplier.

Then, taking the partial derivatives of $\mathcal{L}$ w.r.t w_p and β, and by setting the partial derivatives to zero, we obtain the optimal solutions to w_p according to the SOM algorithms, which are shown in Table 9.

Remark Equation (28) allows us to conclude that at the end of the training of MBSOM-MM**dd** algorithm, when $h_{f(e_k),r} \sim 0$ for $f(e_k) \neq r$, the lower the $\sum_{r=1}^{C} \sum_{e_k \in P_r} \sum_{e \in G_r} d_p(e_k, e)$ the higher the relevance weight w_p of the dissimilarity matrix $\mathbf{D}_p$. Similar remarks can be achieved for MBSOM-MM**dd** and MRBSOM algorithms.

Table 8 Computation of the global weights: Lagrangian functions

SOM algorithms	Lagrangian functions
MBSOM-CM**dd** (Dantas & Carvalho, 2011)	$\mathcal{L} = \sum_{k=1}^{N} \sum_{r=1}^{C} h_{f(e_k),r} \sum_{p=1}^{P} w_p \sum_{e \in G_r} d_p(e_k, e) - \beta\left(\prod_{p=1}^{P} w_p - 1\right).$ (25)
MBSOM-MM**dd**	$\mathcal{L} = \sum_{k=1}^{N} \sum_{r=1}^{C} h_{f(e_k),r} \sum_{p=1}^{P} w_p \sum_{j=1}^{N} (v_{rj})^n d_p(e_k, e_j) - \beta\left(\prod_{p=1}^{P} w_p - 1\right).$ (26)
MRBSOM	$\mathcal{L} = \sum_{k=1}^{N} \sum_{r=1}^{C} h_{f(e_k),r} \sum_{p=1}^{P} w_p \left([\mathbf{D}_p \boldsymbol{\alpha}_r]_k - \frac{1}{2}\boldsymbol{\alpha}_r^\top \mathbf{D}_p \boldsymbol{\alpha}_r\right) - \beta\left(\prod_{p=1}^{P} w_p - 1\right).$ (27)

Table 9 Optimal global weights

SOM algorithms	Global weights
MBSOM-CM**dd** (Dantas & Carvalho, 2011)	$w_p = \frac{\left\{\prod_{s=1}^{P}\left[\sum_{k=1}^{N} \sum_{r=1}^{C} h_{f(e_k),r} \sum_{e \in G_r} d_s(e_k, e)\right]\right\}^{\frac{1}{P}}}{\sum_{k=1}^{N} \sum_{r=1}^{C} h_{f(e_k),r} \sum_{e \in G_r} d_p(e_k, e)}.$ (28)
MBSOM-MM**dd**	$w_p = \frac{\left\{\prod_{s=1}^{P}\left[\sum_{k=1}^{N} \sum_{r=1}^{C} h_{f(e_k),r} \sum_{j=1}^{N} (v_{rj})^n d_s(e_k, e_j)\right]\right\}^{\frac{1}{P}}}{\sum_{k=1}^{N} \sum_{r=1}^{C} h_{f(e_k),r} \sum_{j=1}^{N} (v_{rj})^n d_p(e_k, e_j)}.$ (29)
MRBSOM	$w_p = \frac{\left\{\prod_{s=1}^{P}\left[\sum_{k=1}^{N} \sum_{r=1}^{C} h_{f(e_k),r} \left([\mathbf{D}_s \boldsymbol{\alpha}_r]_k - \frac{1}{2}\boldsymbol{\alpha}_r^\top \mathbf{D}_s \boldsymbol{\alpha}_r\right)\right]\right\}^{\frac{1}{P}}}{\sum_{k=1}^{N} \sum_{r=1}^{C} h_{f(e_k),r} \left([\mathbf{D}_p \boldsymbol{\alpha}_r]_k - \frac{1}{2}\boldsymbol{\alpha}_r^\top \mathbf{D}_p \boldsymbol{\alpha}_r\right)}.$ (30)

Table 10 Computation of the local weights: Lagrangian functions

SOM algorithms	Lagrangian functions
MBSOM-CM**dd** (Dantas & Carvalho, 2011)	$\mathcal{L} = \sum_{k=1}^{N} \sum_{r=1}^{C} h_{f(e_k),r} \sum_{p=1}^{P} w_{rp} \sum_{e \in G_r} d_p(e_k, e) - \sum_{r=1}^{C} \beta_r\left(\prod_{p=1}^{P} w_{rp} - 1\right).$ (31)
MBSOM-MM**dd**	$\mathcal{L} = \sum_{k=1}^{N} \sum_{r=1}^{C} h_{f(e_k),r} \sum_{p=1}^{P} w_{rp} \sum_{j=1}^{N} (v_{rj})^n d_p(e_k, e_j) - \sum_{r=1}^{C} \beta_r\left(\prod_{p=1}^{P} w_{rp} - 1\right).$ (32)
MRBSOM	$\mathcal{L} = \sum_{k=1}^{N} \sum_{r=1}^{C} h_{f(e_k),r} \sum_{p=1}^{P} w_{rp} \left([\mathbf{D}_p \boldsymbol{\alpha}_r]_k - \frac{1}{2}\boldsymbol{\alpha}_r^\top \mathbf{D}_p \boldsymbol{\alpha}_r\right) - \sum_{r=1}^{C} \beta_r\left(\prod_{p=1}^{P} w_{rp} - 1\right).$ (33)

Table 10 provides the Lagrangian functions according to the SOM algorithms for weights assigned locally, where β_r are the Lagrange multipliers.

Then, taking the partial derivatives of $\mathcal{L}$ w.r.t w_{rp} and β_r, and by setting the partial derivatives to zero, we obtain the optimal solutions to w_{rp}, which are shown in Table 11.

Remark Equation (34) allows us to conclude that at the end of the training of MBSOM-CM**dd** algorithm, when $h_{f(e_k),r} \sim 0$ for $f(e_k) \neq r$, the lower the $\sum_{e_k \in P_r} \sum_{e \in G_r} d_p(e_k, e)$ the higher the relevance weight w_{rp} of the dissimilarity matrix $\mathbf{D}_p$ on the cluster P_r. Similar remarks can be achieved for the MBSOM-MM**dd** and MRBSOM algorithms.

2.2.3 Assignment step

During the assignment step, the cluster representatives and the matrix of relevance weights $\mathbf{W}$ are kept fixed. The aim is to minimize the error functions of Table 1 regarding the partition $\mathcal{P}$. Regarding the MBSOM-CM**dd** algorithm, according to Eq. (2), the error function $J_{MBSOM-MMdd}$ is minimized if $\Delta_{\mathbf{W}}(e_k, G_{f(e_k)})$ is minimized for each

Table 11 Optimal local weights

SOM algorithms	Local Weights
MBSOM-CM**dd** (Dantas & Carvalho, 2011)	$w_{rp} = \frac{\left\{\prod_{s=1}^{P}\left[\sum_{k=1}^{N} h_{f(e_k),r} \sum_{e\in G_r} d_s(e_k,e)\right]\right\}^{\frac{1}{P}}}{\sum_{k=1}^{N} h_{f(e_k),r} \sum_{e\in G_r} d_p(e_k,e)}.$ (34)
MBSOM-MM**dd**	$w_{rp} = \frac{\left\{\prod_{s=1}^{P}\left[\sum_{k=1}^{N} h_{f(e_k),r} \sum_{j=1}^{N}(v_{rj})^n d_s(e_k,e_j)\right]\right\}^{\frac{1}{P}}}{\sum_{k=1}^{N} h_{f(e_k),r} \sum_{j=1}^{N}(v_{rj})^n d_p(e_k,e_j)}.$ (35)
MRBSOM	$w_{rp} = \frac{\left\{\prod_{s=1}^{P}\left[\sum_{k=1}^{N} h_{f(e_k),r}\left([\mathbf{D}_s\boldsymbol{\alpha}_r]_k - \frac{1}{2}\boldsymbol{\alpha}_r^\top\mathbf{D}_s\boldsymbol{\alpha}_r\right)\right]\right\}^{\frac{1}{P}}}{\sum_{k=1}^{N} h_{f(e_k),r}\left([\mathbf{D}_p\boldsymbol{\alpha}_r]_k - \frac{1}{2}\boldsymbol{\alpha}_r^\top\mathbf{D}_p\boldsymbol{\alpha}_r\right)}.$ (36)

$e_k \in E$. For a fixed vector of prototypes $\mathcal{P}$ and a fixed matrix of relevance weights $\mathbf{W}$, $\Delta_{\mathbf{W}}(e_k, G_{f(e_k)})$ is minimized if $f(e_k) = arg\min_{1\le s\le C} \Delta_{\mathbf{W}}(e_k, G_s)$.

Following a similar reasoning to MBSOM-MM**dd** and MRBSOM algorithms, Table 12 provides the update rule for the clusters P_r $(1 \le r \le C)$ according to the SOM algorithms and relevance weights of the dissimilarity matrices.

Remark Note that according to Eqs. 37 and 42, the objects are assigned to clusters by computing the dissimilarity between them and the cluster representatives. Dissimilarity matrices that have a big (small) weight strongly (weakly) contribute to computing the dissimilarity between objects and cluster representatives. This is why a dissimilarity matrix with a big weight is more relevant to the clustering task than a dissimilarity matrix with a small weight.

These three steps are repeated until the fixed number of iterations N_{iter} (epochs) is achieved. Algorithm 1 summarizes these steps.

Remark

1. The performance of these SOM algorithms at the end of the training, and the associated partition, depend on the choice of their parameters N_{iter}, σ_0, σ_f, q (MBSOM-CM**dd** family), and n (MBSOM-CM**dd** family);

Table 12 Update rules for the clusters

SOM algorithms	Global weights
MBSOM-CM**dd** (Dantas & Carvalho, 2011)	$P_r = \{e_k \in E : r = f(e_k) = arg\min_{1\le s\le C} \sum_{m=1}^{C} h_{s,m} \sum_{p=1}^{P} w_p D_p(e_k, G_m)\}.$ (37)
MBSOM-MM**dd**	$P_r = \{e_k \in E : r = f(e_k) = arg\min_{1\le s\le C} \sum_{m=1}^{C} h_{s,m} \sum_{p=1}^{P} w_p D_p(e_k, \mathbf{v}_m)\}.$ (38)
MRBSOM	$P_r = \{e_k \in E : r = f(e_k) = arg\min_{1\le s\le C} \sum_{m=1}^{C} h_{s,m} \sum_{p=1}^{P} w_p D_p(e_k, \boldsymbol{\alpha}_m)\}.$ (39)
SOM algorithms	**Local weights**
MBSOM-CM**dd** (Dantas & Carvalho, 2011)	$P_r = \{e_k \in E : r = f(e_k) = arg\min_{1\le s\le C} \sum_{m=1}^{C} h_{s,m} \sum_{p=1}^{P} w_{mp} D_p(e_k, G_m)\}.$ (40)
MBSOM-MM**dd**	$P_r = \{e_k \in E : r = f(e_k) = arg\min_{1\le s\le C} \sum_{m=1}^{C} h_{s,m} \sum_{p=1}^{P} w_{mp} D_p(e_k, \mathbf{v}_m)\}.$ (41)
MRBSOM	$P_r = \{e_k \in E : r = f(e_k) = arg\min_{1\le s\le C} \sum_{m=1}^{C} h_{s,m} \sum_{p=1}^{P} w_{mp} D_p(e_k, \boldsymbol{\alpha}_m)\}.$ (42)

2. For a fixed radius, at each step of the algorithms (representation, weighting, and assignment), the objective functions are locally minimized and decreased. Since the radius is not a variable of the objective functions, its update cannot guarantee that the objective functions decrease regarding the last step before the radius updating; therefore the convergences of algorithms cannot be ensured.
3. Furthermore, the final solution depends on the choice of the C initial cluster representatives (e.g., the C set-medoids, regarding the MBSOM-CM**dd** algorithm).

Algorithm 2 MBSOM-CM**dd**, MBSOM-MM**dd**, and MRBSOM algorithms

```
Require: :
1: The dissimilarity matrices $\mathbf{D}_p\,(1 \leq p \leq P)$, the size map/ , and the number $C$ of neurons (clusters).
2: The distance matrix between the nodes of the grid $\boldsymbol{\delta} = (\|a_s - a_r\|^2)\,(1 \leq s \leq C; 1 \leq r \leq C)$;
3: The number of iterations $N_{iter}$; the initial radius $\sigma_0$; the final radius $\sigma_f$; the parameter $q$ (MBSOM-CMdd); the parameter $n$ (MBSOM-MMdd)
Ensure: : The SOM map; the vector of set-medoids $\mathcal{G}$ (MBSOM-CMdd); the matrix $\mathbf{V}$ of prototype weights (MBSOM-MMdd); the matrix $\mathcal{A}$ of prototype coefficients (MRBSOM); the matrix of relevance weights $\mathbf{W}$; the partition $\mathcal{P}$ of $E$ into $C$ clusters
4: Initialization
5: Set: $t = 0$; $\sigma_{(t)} = \sigma_0 \left(\frac{\sigma_f}{\sigma_0}\right)^{\frac{t}{N_{iter}}}$; $h_{s,r} = \exp\{-\frac{\|a_s - a_r\|2}{2\sigma^2_{(t)}}\}\,(1 \leq s \leq C; 1 \leq R \leq C)$
6: Initial cluster representatives
7: MBSOM-CMdd: randomly select $C$ distinct set-medoids $G_r^{(t)} \in E^{(q)}\,(1 \leq r \leq C)$ to obtain the initial vector of set-medoids $\mathcal{G}^{(t)} = (G_1^{(t)}, \ldots, G_C^{(t)})$.
8: MBSOM-MMdd: randomly initialize the matrix $\mathbf{V}^{(t)} = (v_{rj}^{(t)})_{\substack{1 \leq r \leq C \\ 1 \leq j \leq N}}$ such that $\sum_{j=1}^{N} v_{rj}^{(t)} = 1$ and $v_{rj}^{(t)} \geq 0\,(1 \leq r \leq C)$.
9: MRBSOM: randomly initialize the matrix $\mathcal{A}^{(t)} = (\alpha_{rk}^{(t)})_{\substack{1 \leq r \leq C \\ 1 \leq k \leq N}}$ such that $\sum_{k=1}^{N} \alpha_{rk}^{(t)} = 1$ and $\alpha_{rk}^{(t)} \geq 0\,(1 \leq r \leq C)$.
10: Initial relevance weights
11: Global weights: set $w_p \leftarrow 1\,(1 \leq p \leq P)$. local weights: set $w_{rp} \leftarrow 1\,(1 \leq r \leq C; 1 \leq p \leq P)$
12: Initial assignment:
13: Obtain the initial partition $\mathcal{P}^{(t)} = (P_1^{(t)}, \ldots, P_C^{(t)})$, where $P_r^{(t)}\,(1 \leq r \leq C)$ is computed as
14: MBSOM-CMdd: global weights, according to Eq. (37); local weights, according to Eq. (40).
15: MBSOM-MMdd: global weights, according to Eq. (38); local weights, according to Eq. (41).
16: MRBSOM: global weights, according to Eq. (39); local weights, according to Eq. (42).
17: repeat
18:     Set: $t = t + 1$; $\sigma_{(t)} = \sigma_0 \left(\frac{\sigma_f}{\sigma_0}\right)^{\frac{t}{N_{iter}}}$; $h_{s,r} = \exp\{-\frac{\|a_s - a_r\|2}{2\sigma^2_{(t)}}\}\,(1 \leq s \leq C; 1 \leq R \leq C)$
19:     Step 1: representation:
20:     MBSOM-CMdd: obtain the vector of set-medoids $\mathcal{G}^{(t)} = (G_1^{(t)}, \ldots, G_C^{(t)})$, where the set-medoids $G_r^{(t)} \in E^{(q)}$, representing cluster $P_r$, are computed according to the algorithm (1).
21:     MBSOM-MMdd: compute the components $v_{rj}^{(t)}\,(1 \leq r \leq N; 1 \leq j \leq N)$ of the matrix $\mathbf{V}^{(t)}$ as follows: global weights, according to Eq. (20) and local weights, according to Eq. (21).
22:     MRBSOM: compute the elements of the matrix $\mathcal{A}^{(t)} = (\alpha_{rk}^{(t)})\,(1 \leq r \leq N; 1 \leq k \leq N)$ according to Eq. (24).
23:     Step 2: weighting: compute the relevance weights of the dissimilarity matrices as follows:
24:     MBSOM-CMdd: global weights, according to Eq. (28); local weights, according to Eq. (34).
25:     MBSOM-MMdd: global weights, according to Eq. (29) and local weights, according to Eq. (35).
26:     MRBSOM: global weights, according to Eq. (30) and local weights, according to Eq. (36).
27:     Step 3: assignment:
28:     MBSOM-CMdd: global weights, according to Eq. (37) and local weights, according to Eq. (40).
29:     MBSOM-MMdd: global weights, according to Eq. (38) and local weights, according to Eq. (41).
30:     MRBSOM: global weights, according to Eq. (39) and local weights, according to Eq. (42);
31: until $t = N_{iter}$
```

2.3 The methods

For a simpler presentation of the methods according to the algorithms used and the relevance weights assigned to each dissimilarity matrix, we adopted the following notations.

- MBSOM-CM**dd**-G (Dantas & Carvalho, 2011), if each cluster representative is a set of objects (set-medoids) presenting a matching function described by Eq. (5) and the weights are assigned globally by Eq. (9).
- MBSOM-CM**dd**-L (Dantas & Carvalho, 2011), if each cluster representative is a set of objects (set-medoids) presenting a matching function described by Eq. (5) and the weights are assigned locally by Eq. (12).
- SBSOM-CM**dd**, if the SOM of Ref. Golli et al. (2005) in which each cluster representative is a set of objects (set-medoids).
- MBSOM-MM**dd**-G, if each cluster representative is the entire set of objects weighted on this cluster (weighted medoids) based on Eq. (6) and the weights, are assigned globally by Eq. (10).
- MBSOM-MM**dd**-L, if each cluster representative is the entire set of objects weighted on this cluster (weighted medoids) based on Eq. (6) and the weights are assigned globally by Eq. (13).
- SBSOM-MM**dd**, if the SOM of Ref. Mariño and Carvalho (2020) in which each cluster representative is the entire set of objects weighted on this cluster (weighted medoids).
- MRBSOM-G, if each cluster representative is a normalized linear combination of the objects represented in the description space based on Eq. (7) and the weights are assigned globally by Eq. (11).
- MRBSOM-L, if each cluster representative is a normalized linear combination of the objects represented in the description space based on Equation 7 and the weights are assigned locally by Eq. (14).
- SRBSOM, if the batch SOM of Ref. Hasenfuss and Hammer (2007) in which each cluster representative is a normalized linear combination of the objects represented in the description space.

Note that we also consider related single-view batch SOM algorithms (SBSOM-CM**dd**, SBSOM-MM**dd**, and SRBSOM) in order to compare them with the proposed multiple-view approaches.

2.4 Complexity analysis

The complexity of the algorithms mainly depends on the matching function $\Delta_{\mathbf{W}}$ of Table 2, which allows comparing each object with the representatives of the clusters. Moreover, the final computational complexity considers the following three main steps: representation, weighting, and assignment.

- Regarding the methods MBSOM-CM**dd**-G and MBSOM-CM**dd**-L, the time complexity of Δ is $O(P \times N \times C)$. The representation, weighting (Local or Global), and assignment steps are $O(P \times N^2 \times C)$, $O(P \times N \times C)$), and $O(P \times N \times C^2)$, respectively. The final time complexity is $O(N_{iter} \times P \times N^2 \times C)$.

- In methods MBSOM-MM**dd**-G and MBSOM-MM**dd**-L, the time complexity of Δ is $O(P \times N^2 \times C)$, which is the same in the representation and weighting (local or global) steps, whereas it is $O(P \times N^2 \times C^2)$ for the assignment step. The final time complexity is $O(N_{iter} \times P \times N^2 \times C^2)$.
- Finally, in methods MRBSOM-G and MRBSOM-L, the time complexity of Δ is $O(P \times N^3 \times C)$. In the representation, weighting (local or global), and assignment steps, it is $O(N \times C)$), $O(P \times N^2 \times C)$, and $O(P \times N^3 \times C)$), respectively. The final time complexity is $O(N_{iter} \times P \times N^3 \times C)$.

3 Experimental setting

The successful training of the batch SOM algorithms depends on the choice of their parameters (Badran et al., 2005; Kohonen, 2001, 2013). This section describes the experimental setting used to evaluate the proposed methods compared with other state-of-the-art batch SOM algorithms for multi-view and single-view relational data. The algorithms were implemented in the language "C" and executed on the same machine (OS: Windows 7 64-bits, Memory: 16 GB, Processor: Intel Core i7-X990 CPU @ 3.47 GHz).

A total of 168 distinct experiments were executed by the methods SBSOM-CM**dd**, MBSOM-CM**dd**-L, MBSOM-CM**dd**-G, SBSOM-MM**dd**, MBSOM-MM**dd**-L, and MBSOM-MM**dd**-G. The methods SRBSOM, MRBSOM-L, and MRBSOM-G performed 56 experiments. Experiments proceed with different parameter sets, array maps, shapes, and initials (h_0) kernel values. The search for the best configuration of the proposed methods is based on the trade-off between the internal validity indices *Topographic Error (TE)*, which measure the quality of the SOM map (Kiviluoto, 1996), and the *Silhouette Coefficient (SIL)* (Rousseeuw & Kaufman, 2005), which measures the quality of the cluster partition. The configuration that provides a solution with a not-so-high *TE* and a not-so-low *SIL* is preferred to a configuration that provides a solution with a high *TE* but a low *SIL*. The goal is to obtain a trained map and cluster partition of high quality simultaneously as measured by these indices.

For instance, let us consider the silhouette and topographic error scores of two experiments of one of the assessed methods: experiment 1 ($SIL = 0.70$ and $TE = 0.45$) and experiment 2 ($SIL = 0.68$ and $TE = 0.20$). The setup of experiment 2 is the most suitable according to our methodology. The overall results are presented in the same setting regarding the best behavior in most scenarios, according to the internal indices. This means that if the best results regarding all methods and scenarios are achieved with a squared shape, the methods are compared between them using this configuration with their best setup.

A remarkable feature of multi-view learning is that its performance on an original single-view dataset could still be improved by using manually generated multi-views (Zhao et al., 2017; Sun et al., 2019). Accordingly, datasets that were originally single-view were split into multiple datasets described by disjoint subsets of the original set of features. Fourteen datasets were considered in this study (see Sect. 1 of the Supplementary Material). Table 13 summarizes these datasets, in which P is the number of views, N is the number of objects, M is the number of *a priori* classes, and *ARRAY* is the dimension of the grid array. The maps are arrays of square (Sq.) or rectangle (Rect.) shapes. Table 13 also provides C, the number of neurons (clusters) in the maps for each dataset, which can be deduced from ARRAY. The number of neurons is roughly $\lfloor \sqrt{N} \rfloor$ according to prior research (Vesanto et al., 1999).

For each dataset, the Euclidean distance is used to compute a dissimilarity matrix simultaneously considering all the variables in each view. Regarding IRIS and WINE datasets, for each variable describing the objects a dissimilarity matrix is computed using the Euclidean distance.

Then, the matrices were normalized according to their overall dispersion to have the following dynamic range: each dissimilarity $d(e_k, e_l)\,(1 \leq k, l \leq N)$ in a given dissimilarity matrix $\mathbf{D}$ is normalized as $\frac{d(e_k,e_l)}{T}$, where $T = \sum_{k=1}^{N} d(e_k, g)$ is the overall dispersion and $g = e_l \in E = \{e_1, \ldots, e_N\}$ is the overall representative, which is computed according to $l = arg\min_{1 \leq h \leq N} \sum_{k=1}^{N} d(e_k, e_h)$. The considered batch SOM algorithms operate on these normalized dissimilarity matrices.

Regarding the single-view batch SOM algorithms of Refs. Golli et al. (2005); Hasenfuss and Hammer (2007); Mariño and Carvalho (2020), the single dissimilarity matrix of each dataset is obtained using a concatenation strategy, i.e., as the average of the dissimilarity matrices describing this dataset.

In addition, we consider the Gaussian neighborhood kernel function (Eq. 43) that ranges in [0, 1]:

$$h_{s,r} = \exp\left\{ -\frac{\|a_s - a_r\|^2}{2\sigma_{(t)}^2} \right\}. \tag{43}$$

where $\|a_s - a_r\|^2$ is the squared Euclidean distance in the topological space between neurons a_s and a_r and $\sigma_{(t)}$ is the kernel width (radius) at the iteration t.

The fixed initial value of the kernel width $\sigma_0 = \sigma_{(0)}$ and the final value $\sigma_f = \sigma_{(N_{iter})}$ are updated at each iteration t according to Equation (44). Each method is executed using $N_{iter} = 50$ iterations (cf. Ref. Kohonen et al. (2014)).

$$\sigma_{(t)} = \sigma_0 \left(\frac{\sigma_f}{\sigma_0} \right)^{\frac{t}{N_{iter}}}. \tag{44}$$

About the initial and final values of the kernel width, some heuristic methods proposed in the literature are mainly related to the classical SOM assignment based only on the distance between objects and BMU (Vesanto et al., 1999). In our case, since the assignment is based on the generalized dissimilarity functions (c.f. Table 2), which leads to maps with a high topographic error, we propose to fix the values of σ_0 and σ_f by using the heuristic method of Ref. Carvalho et al. (2022). The initial and final values of the radius σ are computed according to the expressions (45) and (46), respectively. Thus, we initialized the map radius (σ_0) representing the distance of two neurons from a kernel value (h_0) equal to 0.1. In addition, we computed σ_f such that two neighboring neurons have a kernel value (h_f) equal to 0.01. The map diameter in the topological space is the largest topological distance between two neurons of the map and is computed as $(x_{max})^2 + (y_{max})^2$, where x_{max} and y_{max} correspond to the grid size in the horizontal X-axis and vertical Y-axis, respectively.

$$\sigma_0 = \sqrt{\frac{-[(x_{max})^2 + (y_{max})^2]}{2\ln(h_0)}}. \tag{45}$$

Table 13 Summary of the multi-view datasets

Dataset	P	N	M	Array	C
Core 1	7	400	4	3×6	18
Core 2	7	400	4	3×6	18
Reuters	5	1200	6	6×6	36
Dermatology	2	366	6	3×3	9
Aloi cars	4	1413	13	6×6	36
Spectf	2	267	2	3×5	15
Forest type	2	325	4	3×6	18
Water	2	527	13	3×7	21
Image	2	2310	7	7×7	49
Flower 17	4	1360	17	6×6	36
Mfeat	6	2000	10	4×9	36
Phoneme	3	2000	5	6×6	36
Iris	4	150	3	2×4	8
Wine	13	178	3	2×4	8

$$\sigma_f = \sqrt{\frac{-1}{2\ln(h_f)}}. \tag{46}$$

Specifically for the models SBSOM-CM**dd**, MBSOM-CM**dd**-L, and MBSOM-CM**dd**-G, after some trials based on the trade-off between *TE* and *SIL*, the cardinal q of the set-medoids was fixed to 5. In turn, after some trials, the parameter n, which controls the level of smoothness of the distribution of prototype weights among all the objects in each of the clusters, was fixed to 1.1 for the models SBSOM-MM**dd**, MBSOM-MM**dd**-L, and MBSOM-MM**dd**-G based also on the trade-off between *TE* and *SIL*. The experiments were repeated and randomly initialized 10 times. For each loop, 50 iterations were performed.

Finally, to assess the degree of agreement between an *a priori* partition and a partition provided by the SOM algorithms, we used two external validity indices computed on datasets with labeled instances: the *F-measure* (Manning et al., 2008) and the *Normalized Mutual Information (NMI)* (Cover & Thomas, 2006). See Sect. 1 of the supplementary material for further information.

4 Experimental analysis and discussion

This section assesses the performance of the proposed multi-view SOM algorithms, namely MBSOM-MM**dd**-L, MBSOM-MM**dd**-G, MRBSOM-L, and MRBSOM-G compared with already in the literature multi-view SOM algorithms (Dantas & Carvalho, 2011): MBSOM-CM**dd**-L and MBSOM-CM**dd**-G. In addition, we compared them with the most related single-view methods SBSOM-CM**dd** (Golli et al., 2005), SBSOM-MM**dd** (Mariño & Carvalho, 2020), and SRBSOM (Hasenfuss & Hammer, 2007).

The algorithms are compared according to *F-measure*, *NMI*, *TE*, and *SIL*. Table 14 presents the average ranks concerning the compared method in each metric. The average rank provides the criterion for evaluating the models. It means that algorithms that achieve higher mean scores for *F-measure*, *NMI*, or *Silhouette* perform the best, whereas algorithms with lower mean scores for *Topographic error* outperform the others. Regarding average ranks, lower values imply better results. In this table, the best average rank provided by each index is in bold and italic.

According to Table 14, MRBSOM-G had the best performance regarding the average rank of *F-measure*, *NMI*, and *TE*. MBSOM-MM**dd**-L surpassed the other algorithms regarding the average rank of *SIL*. The multi-view variants improved the single-view variants and the proposed algorithms outperformed the multi-view benchmarks (set-medoids SOM) in most cases. The global-weighted algorithms outperformed the local-weighted algorithms regarding *NMI* and *TE*. Local-weight algorithms performed the best concerning *SIL*.

Based on the average ranks, we used the Friedman's test (Friedman, 1940; Demšar, 2006) to test the null hypothesis that all models perform the same regarding *F-measure*, *NMI*, *TE*, and *SIL* indices. The test reached a p-value $= 9.899e^{-17}$ and p-value $= 3.422e^{-17}$ concerning *TE* and *Silhouette* respectively. The obtained p-value allows us to reject the null hypothesis that all the algorithms perform the same concerning these indices. To detect significant pairwise differences among all models, the *Nemenyi*'s test (Nemenyi, 1963; Demšar, 2006) was applied regarding the average rank of *NMI* and *TE*, available in Table 14. For a significance level α, the test determines the critical difference (CD). If the difference between the average ranking of the two algorithms is greater than CD, the null hypothesis that the algorithms have the same performance is rejected.

Figures 1 and 2 show the CD plot to visualize the differences concerning the *TE* and *SIL* indices. Models that are not significantly different (with $\alpha = 0.05$) are connected (Demšar, 2006). The first positions in the figures indicate lower values in the average rank, hence the best algorithm. The critical difference (*CD*) determined by the *Nemenyi* post-test was 3.211.

Concerning *TE*, algorithms MRBSOM-G, MRBSOM-L, and SRBSOM significantly improved algorithms SBSOM-CM**dd** and MBSOM-CM**dd**-L. Local-weight algorithms performed better concerning *SIL*, algorithms MBSOM-MM**dd**-L and MBSOM-CM**dd**-L presented significant improvements concerning SBSOM-MM**dd**, which had the worst performance in this case. Finally, the proposed multi-view SOM algorithms presented a better performance than the already in the literature multi-view SOM algorithms; however, these improvements are not always statistically significant.

Dermatology dataset

Next, we provide more detailed results for the Dermatology dataset (Dua & Graff, 2017) to demonstrate the usefulness of the proposed algorithms. Figure 3 displays the distribution of the objects over 9 nodes on the 3×3 grid provided by the proposed MBSOM-MM**dd**-G algorithm on the Dermatology dataset. Each node represents a cluster (a neuron). In addition, the size of the circle is proportional to the number of objects in the cluster. Also, the total area of the circle is shared between the areas corresponding to the *a priori* classes.

In this figure, the map tends to mix the data of patients diagnosed with *seboreic dermatitis* and *pityriasis rosea*. This might be linked to both diseases presenting very similar manifestations. Similar behavior was observed in all other maps. Furthermore, this map, and overall those produced by the proposed methods, tend to better cluster data from patients previously diagnosed with *pityriasis rubra pilaris*, which are less well represented in the dataset. Finally, compared to the set-medoids SOM algorithms on this dataset, the clusters generated by the proposed MBSOM-MM**dd**-G algorithm (as well as by the other proposed methods) are more homogeneous.

Table 14 Average Rank by index and algorithm

Methods	F-measure	NMI	TE	SIL
SBSOM-CMdd	5.857	6.214	6.714	5.857
MBSOM-CMdd-L	5.571	5.643	6.857	3.571
MBSOM-CMdd-G	5.571	4.964	5.107	4.714
SBSOM-MMdd	5.357	4.929	6.143	7.357
MBSOM-MMdd-L	4.429	5.000	5.214	***3.429***
MBSOM-MMdd-G	4.643	4.643	4.821	4.714
SRBSOM	4.893	5.000	3.464	5.000
MRBSOM-L	4.714	4.571	3.429	4.357
MRBSOM-G	***3.964***	***4.036***	***3.250***	6.000

If the *a priori* number of classes corresponds to the true and unknown number of clusters, the proposed SOM algorithms, even when they run with a number of neurons (clusters) larger than the number of clusters (classes), tend to be able to find the true number of clusters. Accordingly, the proposed algorithms tend to leave some clusters empty, and thus produce a better mapping, from a clustering point of view. The empty nodes help separate the main groups of diseases. For more details, Sect. 4 of the Supplementary Material describes this dataset and presents the learned maps provided by the SOM algorithms.

Table 15 shows the results for the internal and external indices computed for the solution with the best objective function value over the 10 executions for the Dermatology dataset. In this table, regarding each metric, the best score is highlighted in bold; the result is followed by the rank, which is shown in italic; and the best rank among all methods is shown in bold and italic.

Concerning *F-measure* and *NMI*, the multi-medoids SOM algorithms had the best performance and the relational SOM algorithms had the second best performance. Moreover, the set-medoids SOM algorithms had the worst performance.

Overall, the multi-view algorithms outperformed the single-view algorithms within the same family, except for MBSOM-MM**dd**-G, which is outperformed by SBSOM-MM**dd** in terms of *F-measure*, and MRBSOM-L, which is outperformed by SRBSOM in terms of *NMI*. Regarding *TE* and *SIL*, the MRBSOM-L and MRBSOM-G models achieved better results than the SRBSOM model. MBSOM-MM**dd**-L and MBSOM-MM**dd**-G outperformed SBSOM-MM**dd** concerning *SIL* index. MBSOM-CM**dd**-L and MBSOM-CM**dd**-G overcame SBSOM-CM**dd** regarding *SIL* index. Finally, the algorithms with the worst overall performance were SBSOM-CM**dd**, MBSOM-CM**dd**-L, and MBSOM-CM**dd**-G. The proposed methods with the best performance were MBSOM-MM**dd**-L and MRBSOM-G.

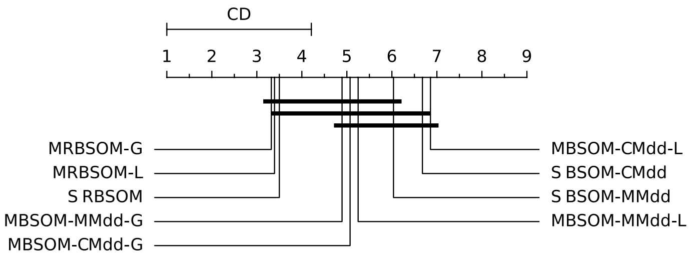

Fig. 1 Comparison of all models against each other through the Nemenyi test according to *TE* with $\alpha = 0.05$

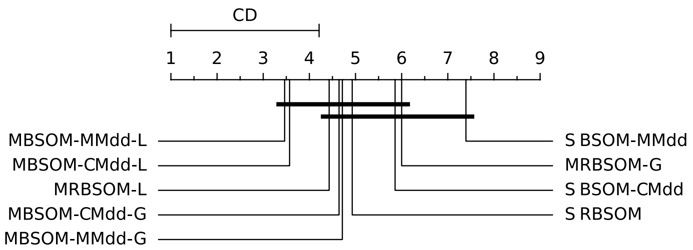

Fig. 2 Comparison of all models against each other through the Nemenyi test according to *SIL* with $\alpha = 0.05$

In conclusion, regarding the overall mean ranking, the multi-view algorithms outperformed the single-view algorithms. The multi-medoids and relational SOM surpassed the set-medoids SOM.

For illustrative purposes, Table 16 shows the vectors of relevance weights locally and globally estimated for the solution with the best objective function value over the 10 executions for the Dermatology dataset. The view with the highest relevance to each of the globally weighted approaches and the most relevant view for the clusters concerning each one of the locally weighted approaches are highlighted in boldface in this table. Among all methods, View 2 had the greatest influence on the outputs. However, concerning the local-weighted algorithms, View 1 had the greatest influence on defining clusters 4 and 6 produced by the MBSOM-CM**dd**-L, as well as on defining cluster 2 produced by the MBSOM-MM**dd**-L and clusters 1 and 2 produced by the MRBSOM-L algorithm.

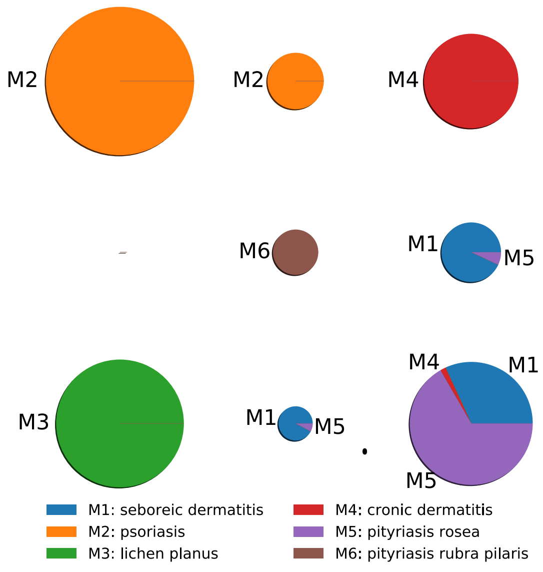

Fig. 3 SOM maps provided by the proposed method MBSOM-MM**dd**-G for the Dermatology dataset

5 Final remarks

The paper proposes two new families of batch SOM algorithms for multi-view dissimilarity data: multi-medoids SOM and relational SOM. Both families of algorithms are designed to provide a crisp partition to learn the relevance weight for each dissimilarity matrix and provide a representative for each cluster based on a suitable objective function. The goal of the proposed algorithms is to preserve the topological properties of the data on the map. The relevance of each dissimilarity matrix can be computed locally for each cluster in MBSOM-MM**dd**-L and MRBSOM-L methods, or globally for the whole partition in MBSOM-MM**dd**-G and MRBSOM-G methods. MBSOM-MM**dd**-L, and MBSOM-MM**dd**-G consider the cluster representatives as vectors of weights whose components measure how the objects are weighted as a medoid in a given cluster. In MRBSOM-L and MRBSOM-G, each cluster representative is a normalized linear combination of the objects represented in the description space.

Experiments with 14 datasets were performed by means of similar parametrizations for an efficient evaluation. The comparison was established using the multi-view SOM algorithms MBSOM-CM**dd**-G and MBSOM-CM**dd**-L (Dantas & Carvalho, 2011), as well as the single-view SOM algorithms SBSOM-CM**dd** (Golli et al., 2005), SRBSOM (Hasenfuss & Hammer, 2007), and SBSOM-MM**dd** (Mariño & Carvalho, 2020).

We tested the null hypothesis that all models perform the same and the non-parametric Friedman's test together with the *Nemenyi* post-test indicated merely random differences in terms of the performance measures.

Our findings indicate that the proposed models presented better performance than the benchmarks, although not necessarily statistically significant improvements were obtained. In most cases, MRBSOM-G, MRBSOM-L, SRBSOM (Hasenfuss & Hammer, 2007), MBSOM-MM**dd**-G, and MBSOM-MM**dd**-L outperformed the other methods. Specifically, methods MRBSOM-G, MRBSOM-L, and SRBSOM (Hasenfuss & Hammer, 2007) significantly improved methods SBSOM-CM**dd** (Golli et al., 2005), and MBSOM-CM**dd**-L concerning *TE*. In addition, we provide more detailed results for the Dermatology dataset in which the proposed methods were the best-ranked.

In conclusion, the multi-view models outperformed the single-view models. Moreover, the multi-medoids and relational SOM algorithms performed better than the set-medoids SOM algorithm. In further research, we aim to extend this study considering prior and new on-line versions of the proposed SOM algorithms.

Table 15 Results for the Dermatology dataset

Methods	F-measure		NMI		TE		SIL	
SBSOM-CMdd	0.716	*9*	0.680	*9*	0.246	*7*	0.234	*8*
MBSOM-CMdd-L	0.755	*8*	0.704	*8*	0.331	*8*	0.244	*6*
MBSOM-CMdd-G	0.813	*7*	0.798	*7*	0.454	*9*	0.235	*7*
SBSOM-MMdd	0.876	*2*	0.832	*5*	0.137	*3*	0.233	*9*
MBSOM-MMdd-L	**0.891**	***1***	**0.857**	***1***	0.161	*4*	**0.316**	***1***
MBSOM-MMdd-G	0.861	*3*	0.846	*3*	0.230	*6*	0.283	*4*
SRBSOM	0.845	*6*	0.840	*4*	0.199	*5*	0.252	*5*
MRBSOM-L	0.851	*5*	0.825	*6*	0.126	*2*	0.308	*3*
MRBSOM-G	0.856	*4*	0.847	*2*	**0.060**	***1***	0.312	*2*

Table 16 Dermatology dataset: vectors of relevance weights

Cluster	MBSOM-CMdd-L		MBSOM-MMdd-L		MRBSOM-L	
	View 1	View 2	View 1	View 2	View 1	View 2
1	0.805	**1.242**	0.893	**1.119**	**1.001**	0.999
2	0.961	**1.041**	**1.057**	0.946	**1.029**	0.972
3	0.907	**1.102**	0.833	**1.200**	0.887	**1.127**
4	**1.300**	0.769	0.993	**1.007**	0.953	**1.049**
5	0.886	**1.129**	0.937	**1.067**	0.984	**1.016**
6	**1.022**	0.978	0.983	**1.017**	0.973	**1.028**
7	0.907	**1.102**	0.888	**1.126**	0.985	**1.016**
8	0.968	**1.033**	0.925	**1.081**	0.971	**1.030**
9	0.990	**1.010**	0.880	**1.137**	0.992	**1.008**
	MBSOM-MMdd-G		MBSOM-MMdd-G		MRBSOM-G	
	0.920	**1.087**	0.896	**1.116**	0.977	**1.024**

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s10994-024-06607-x.

Acknowledgements The authors would like to thank the IFCS 2022 MLJ special issue guest editors and the anonymous referees for their careful revision, the Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq (311164/2020-0), and the Fundação de Amparo à Ciência e Tecnologia do Estado de Pernambuco - FACEPE (IBPG-0820-1.03/19), for partially financing this study.

Author contributions Laura María Palomino Mariño: study design, methodology, software, writing, review, and editing; Francisco de Assis Tenorio de Carvalho: supervision, study design, methodology, writing, review, and editing.

Funding This paper was partially supported by the Brazilian Agencies Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq (311164/2020-0) and the Fundação de Amparo à Ciência e Tecnologia do Estado de Pernambuco - FACEPE (IBPG-0820-1.03/19).

Availability of data and materials The data are available upon request to the authors.

Code availability The code is available upon request to the authors.

Declarations

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

Ethical approval Not applicable.

Consent to participate Not applicable.

Consent for publication Not applicable.

References

Astudillo, C. A., & Oommen, B. J. (2014). Topology-oriented self-organizing maps: A survey. *Pattern Analysis and Applications, 17*(2), 223–248.

Badran, F., Yacoub, M., & Thiria, S. (2005). Self-organizing maps and unsupervised classification. In G. Dreyfus (Ed.), *Neural networks* (pp. 379–442). Springer.

Carvalho, F., Irpino, A., Verde, R., & Balzanella, A. (2022). Batch self-organizing maps for distributional data with an automatic weighting of variables and components. *Journal of Classification, 39*(2), 343–375.

Cleuziou, G., Exbrayat, M., Martin, L., & Sublemontier, J.-H. (2009). Cofkm: A centralized method for multiple-view clustering. In: *2009 Ninth IEEE International Conference on Data Mining*, (pp. 752–757). IEEE.

Cottrell, M., Olteanu, M., Rossi, F., & Villa-Vialaneix, N. (2018). Self-organizing maps, theory and applications. *Revista Investigatión Operational, 39*(1), 1–22.

Cover, T. M., & Thomas, J. A. (2006). *Elements of information theory* (2nd ed.). Wiley.

Dantas, A. B. S., & Carvalho, F. A. T. (2011). Adaptive batch som for multiple dissimilarity data tables. In: *2011 IEEE 23rd International Conference on Tools with Artificial Intelligence* (pp. 575–578). IEEE.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research, 7*, 1–30.

Diday, E., & Govaert, G. (1977). Classification automatique avec des distances adaptatives. *R.A.I.R.O. Informatique Computer Science, 11*(4), 329–349.

Domínguez-González, M., Fuertes-Martínez, J. J., Blanco, I. D., Prada, M. A., Alonso, S., & Álvarez, A. M. (2012). Monitoring industrial processes with som-based dissimilarity maps. *Expert Systems with Applications, 39*(8), 7110–7120.

Douzas, G., Rauch, R., & Bação, F. (2021). G-SOMO: An oversampling approach based on self-organized maps and geometric SMOTE. *Expert Systems with Applications, 183*, 115230.

Dua, D., & Graff, C. (2017). UCI Machine Learning Repository. http://archive.ics.uci.edu/ml

Friedman, M. (1940). A comparison of alternative tests of significance for the problem of m rankings. *The Annals of Mathematical Statistics, 11*(1), 86–92.

Frigui, H., Hwang, C., & Rhee, F. C. (2007). Clustering and aggregation of relational data with applications to image database categorization. *Pattern Recognition, 40*(11), 3053–3068.

Golli, A. E., Conan-Guez, B., & Rossi, F. (2005). A self-organizing map for dissimilarity data. In: D.B. Al (Ed.) *Classification, Clustering, and Data Mining Applications – Proceedings of the Meeting of the International Federation of Classification Societies (IFCS), Chicago, USA, 15–18 July 2004* (pp. 61–68). Springer.

Gusmão, R. P., & Carvalho, F. A. T. (2019). Clustering of multi-view relational data based on particle swarm optimization. *Expert Systems with Applications, 123*, 34–53.

Hammer, B., & Hasenfuss, A. (2007). Relational neural gas. In: J. Hertzberg, M. Beetz, & R. Englert (Eds.) *Advances in Artificial Intelligence*. KI 2007. Lecture Notes in Computer Science, (pp. 190–204). Springer.

Hasenfuss, A., & Hammer, B. (2007). Relational topographic maps. In: M. Berthold, J. Shawe-Taylor, & N. Lavrac, (Eds.) *Advances in intelligent data analysis VII. IDA 2007*. Lecture Notes in Computer Science, (vol. 4723, pp. 93–105). Springer.

Huang, J. Z., Ng, M. K., Rong, H., & Li, Z. (2005). Automated variable weighting in k-means type clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 27*(5), 657–668.

Kamimura, R. (2019). Som based information maximization to improve and interpret multi-layered neural networks: From information reduction to information augmentation approach to create new information. *Expert Systems with Applications, 125*, 397–411.

Kaski, S., Kangas, J., & Kohonen, T. (1998). Bibliography of self-organizing map (som) papers: 1981–1997. *Neural Computing Surveys, 1*(3–4), 1–176.

Kaufman, L. S., & Rousseeuw, P. J. (1987). Clustering by means of medoids. Tatistical data analysis based on the l1 norm. Y. Dodge (pp. 405–416).

Kiviluoto, K. (1996). Topology preservation in self-organizing maps. In: *Proceedings of International Conference on Neural Networks (ICNN'96), Washington, DC, USA, June 3–6, 1996,* (pp. 294–299).

Kohonen, T. (2014). Matlab implementations and applications of the self-organizing map. *Unigrafia Oy, Helsinki,177*.

Kohonen, T. (2001). *Self-organizing maps* Springer Series in Information Sciences (3rd ed.). Springer.

Kohonen, T. (2013). Essentials of the self-organizing map. *Neural Networks, 37*, 52–65.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. Cambridge University Press.

Mariño, L. M. P., & Carvalho, F. A. T. (2020). A new batch SOM algorithm for relational data with weighted medoids. IJCNN 2020, Glasgow, United Kingdom, July 19–24, 2020*2020 International Joint Conference on Neural Networks* (pp. 1–8). IEEE.

Nemenyi, P. B. (1963). *Distribution-free multiple comparisons*. Princeton University.

Oja, M., Kaski, S., & Kohonen, T. (2003). Bibliography of self-organizing map (som) papers: 1998–2001 addendum. *Neural Computing Surveys, 3*(1), 1–156.

Olteanu, M., & Villa-Vialaneix, N. (2015). On-line relational and multiple relational som. *Neurocomputing, 147*, 15–30.

Olteanu, M., Villa-Vialaneix, N., & Cottrell, M. (2012). On-line relational SOM for dissimilarity data. Advances in Intelligent Systems and ComputingIn P. A. Estévez, J. C. Príncipe, & P. Zegers (Eds.), *Advances in self-organizing maps - 9th International Workshop, WSOM 2012, Santiago, Chile, December 12–14, 2012, Proceedings* (Vol. 198, pp. 13–22). Springer.

Rousseeuw, P. J., & Kaufman, L. (2005). *Finding groups in data: An introduction to cluster analysis*. Wiley.

Sun, S., Mao, L., Dong, Z., & Wu, L. (2019). *Multiview machine learning*. Springer.

Vatanen, T., Osmala, M., Raiko, T., Lagus, K., Sysi-Aho, M., Orešič, M., Honkela, T., & Lähdesmäki, H. (2015). Self-organization and missing values in som and gtm. *Neurocomputing, 147*, 60–70.

Vesanto, J., Himberg, J., Alhoniemi, E., & Parhankangas, J. (1999). Self-organizing map in matlab: The som toolbox. *Proceedings of the Matlab DSP Conference, 99*, 16–17.

Wang, Q., Lv, H., Yue, J., & Mitchell, E. (2017). Supervised multiview learning based on simultaneous learning of multiview intact and single view classifier. *Neural Computing and Applications, 28*(8), 2293–2301.

Xu, C., Tao, D., & Xu, C. (2015). Multi-view intact space learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 37*(12), 2531–2544.

Yang, Y., & Wang, H. (2018). Multi-view clustering: A survey. *Big Data Mining and Analytics, 1*(2), 83–107.

Zhao, J., Xie, X., Xu, X., & Sun, S. (2017). Multi-view learning overview: Recent progress and new challenges. *Information Fusion, 38*, 43–54.