



Subspace Clustering Multi-module Self-organizing Maps with Two-Stage Learning

Marcondes R. da Silva Júnior^(✉)  and Aluizio F. R. Araújo 

Center of Informatics, Federal University of Pernambuco, Recife, Brazil
{mrsj, aluizioa}@cin.ufpe.br

Abstract. Clustering complexity increases with the number of categories and sub-categories and with data dimensionality. In this case, the distance metrics lose discrimination power with the growth of such dimensionality. Thus, we propose a multiple-module soft subspace clustering algorithm called Subspace Clustering Multi-Module Self-Organizing Maps (SC-MuSOM) that produces a map for each category. Moreover, SC-MuSOM learns a relevance coefficient for each dimension of each cluster handling the dimensionality curse. This fast-training model has a second learning stage in which the cluster prototypes are finely tuned considering the spatial resemblance between cluster centers. We validated the model with data mining sets from UCI Repository and computer vision data. Our experiments suggest that SC-MuSOM is competitive with other state-of-the-art models for the tested problems.

Keywords: Subspace clustering · High-dimensional data · Self-organizing maps · Multiple-model clustering · Fine-tuning stage

1 Introduction

High-dimensional data originating from images, videos, and texts have been increasingly used in machine learning. With the increasing availability of cheap storage and the rising use of sensor data, we notice substantial data growth in both volume and dimensionality. Data sets with many attributes are referred to as high dimensional. Their complexity turn them hard to handle, therefore, suitable computational tools are needed to process such type of data [1].

An important difficulty in the categorization of high-dimensional data is the progressive reduction of the discriminant capacity of dissimilarity distances caused by the growth of dimensionality [2], therefore data can be miscategorized. Thus, previous works accomplished dimensionality reduction by creating new dimensions through combinations of the original attributes. Such a dimensionality decrease provokes loss of information, even yet preserving information about irrelevant dimensions [3, 4].

We thank FACEPE (Fundação de Amparo à Ciência e Tecnologia do Estado de Pernambuco) for financial support project.

We propose a simple solution, called Subspace Clustering Multi-Module Self-Organizing Maps (SC-MuSOM), that can effectively cluster high-dimensional data reducing the curse dimensionality problem through decomposition of the learning process and use of dimension relevance for identifying the more significant attributes for each cluster. Moreover, such a model has a significantly lower computational cost when compared with state-of-art models.

Our model is made up of Self-organizing Maps (SOM) modules. The model design entails a multi-module architecture in which each module is trained only with instances of the same category. In spite of having a number of modules, this is not an assembly model, hence, the most active unit of all modules determines the category of an input pattern. In the training stage, a relevance vector associated with each node is updated at the end of each epoch. Such relevancies weigh the importance of each dimension as a discriminant attribute for a cluster. Therefore, the most discriminating features for each category count more on the distance metrics than on the less relevant attributes. Furthermore, SC-MuSOM has a second learning stage that aims to reduce the miscategorization of some patterns due to spatial proximity between prototypes of different categories. A mutual moving-away movement of close prototypes typifying different clusters is carried out to reduce the odds of errors.

In sum, the proposed model differs from the original SOM mainly in:

- Use of multiple modules to reduce the losses of the curse of dimensionality;
- Use of relevance to weigh each attribute in the distance calculation;
- Use of a refinement stage moving apart prototypes of different categories;
- Definition of a second learning stage for refinement of the clustering method.

In Sect. 2, the subspace clustering and learning refinement are defined and some papers on them are briefly discussed. Section 3 presents SC-MuSOM. In Sect. 4, we show the experimental results and their analysis. Finally, Sect. 5 concludes the paper and makes some suggestions for future research.

2 Self-organizing Maps and High-Dimensional Data Clustering

This section briefly discusses clustering for high-dimensional data, algorithms based on SOM for categorization of that type of data, and, subspace clustering and methods for learning refinement.

2.1 Clustering of High-Dimensional Data

Clustering consists of the process of partitioning unlabeled datasets into groups of similar elements. Each cluster comprises patterns more similar to one another according to a given measure than patterns belonging to other groups [2]. Data clustering is a technique useful in many types of problems such as data mining in which solid knowledge can be extracted from apparently unstructured data. When analyzing data with high-dimensionality, data having a few dozens to

many thousands of dimensions, the ordinary distance measures often lose their discriminative capacity, thus, clustering performance is negatively impacted by that. As a consequence, clustering methods based on traditional distance measures often fail to achieve acceptable performance since many feature dimensions may be irrelevant or redundant to certain clusters [2].

Pioneer methods reduce the data dimensionality or use a parsimonious Gaussian mixture model (GMM) to overcome the curse of dimensionality. Very often, the reduction of dimensionality is followed by a classical clustering method. Dimension reduction techniques typically employ feature extraction and feature selection for generate new variables preserving most of the global information. The parsimonious models require the estimation of some parameters. Such models allow clustering methods based on the expectation-maximization.

Subspace Clustering (SC) algorithms identify different subspaces to determine distinct clusters. Their two main types are Hard Subspace Clustering (HSC) and Soft Subspace Clustering (SSC) [5]. HSC algorithms, identify the exact subspaces for different clusters and assign 1 or zero to each relevance. In turn, SSC algorithms perform clustering by assigning a real-valued weight to each feature (attribute) of each cluster to weigh its importance to that particular cluster [2]. Such algorithms are classified as conventional SSC (CSSC), independent SSC (ISSC), and extended SSC (XSSC). In CSSC, all clusters share the same subspace and a common weight vector. The weight vectors in ISSC are determined independently for each cluster. XSSC was developed by extending CSSC or ISSC algorithms for some specific purposes or introducing new mechanisms to enhance clustering performance [5].

2.2 Self-organizing Maps for Subspace Clustering

The SOM and LVQ are prototype-based neural network models. They use a distance measure to select the cluster prototypes to be updated. Euclidean distance tends to lose its discriminating power as the number of attributes increases. So, there are LVQ and SOM-based models that assign a weight, the relevance [6], to each attribute differentiating the importance of each dimension for each cluster. The proposed algorithms obtain relevance attributes in the learning process.

Some recent LVQ models were recently proposed for subspace clustering. GMLVQ and GLVQ had their activation function step modified for improving the categorization process [7]. Intuitive indications of feature relevancies for a given model's decision can be provided by metric-learning approaches, such as GRLVQ [6] that adapts a diagonal matrix, scaling the relevance of the input features. Generalizations that use a full matrix, such as in GMLVQ [8] exist, however, a single global quadratic matrix remains the most common choice. A few approaches extend this setting to non-global matrices, such as LGRLVQ [9] and LGMLVQ [8]. However, they allow only one matrix per prototype, which corresponds to one metric per Voronoi cell in the input space. The mentioned LVQ-based methods are up-to-date alternatives that assign relevance to attributes to make the distance metrics more discriminating.

Also, some SOM-based approaches were proposed, we can mention Dimension Selective Self-organizing Map (DSSOM) [10], Local Adaptive Receptive Field Dimension Selective Self-organizing Map (LARFDSSOM) [11], Local Adaptive Receptive Field Dimension Selective Self-organizing Map 2 (LARFDSSOM2) [4], and Soft Subspace Neural Gas for Data Stream Clustering (S2G-Stream) [12]. DSSOM can find clusters and identify their relevant dimensions, simultaneously with the self-organizing process. The next models are time-varying structure SOM, TVS-SOM [13]. LARFSSOM is characterized by a structure of nodes that change according to the defined level of resemblance between a pattern and the prototype of the category it belongs to. LARFSSOM2 is a version that improves learning capacity by refining the relevance accuracy. S2G-Stream algorithm is based on growing neural gas. The authors proposed two models of feature and block weighting (global and local), a method for SSC on an evolving data stream. We returned to the original SOM with a fixed structure as a simple solver for the same class of problems.

2.3 Learning Enhancement

Several adaptations were proposed to the original SOM to ameliorate categorization. Some of them were accuracy-oriented for high-dimensional data such as a type of assembly of SOMs [14]. The method connects the output of a SOM to the input of the next SOM. It also utilizes the characteristic of high-dimensional data insensitivity to update the values of dimensions. Another approach is a hierarchical categorization in which patterns that have imprecise categorization can undergo further training with different data composition from initial training to refine a solution [15]. SOM-based model performance can be improved with a second process of weight adjustment. Bi-organization local refinement [16] organizes the data by defining an appropriate representation and metric such that they respect the smoothness and structure underlying the data.

Many changes were proposed to the original SOM to ameliorate categorization. Some of them were accuracy-oriented for high-dimensional data such as a type of assembly of SOMs [14]. In this approach, each SOM is trained with some different criteria and after training all expert SOMs, the output is determined by a criterion considering the specialists. Cascaded SOM is an extension of classical SOM [14] in which multiple SOMs are connected in series, the so-called cascaded structure. The method connects the output of a SOM to the input of the next SOM. It also utilizes the characteristic of high-dimensional data insensitivity to update the values of dimensions.

Another approach is hierarchical categorization, in which patterns that have imprecise categorization can undergo further training with different data compositions from initial training to refine a solution. Hierarchical sparse subspace clustering (HESSC) [15] deals with high-dimensional and highly mixed data robustly and quickly. The algorithm performs a hierarchical subspace clustering and it determines the number of clusters.

SOM-based model performance can be improved with a second process of weight adjustment. Bi-organization local refinement [16] organizes the data by

defining an appropriate representation and metric such that they respect the smoothness and structure underlying the data. They also aim to generalize the joint clustering of observations and features, in the case the data does not fall into clear disjoint groups. The process consists of an iterative refinement procedure that exploits the co-dependencies between features and observations.

The learning process of SC-MuSOM is fully based on SOM. Thus, we train each SOM module with patterns belonging to a particular category. Moreover, the learning process has two stages: the first training stage is based on the original SOM and the second stage modifies locally the Voronoi regions.

3 Subspace Clustering Multi-module Self-organizing Maps (SC-MuSOM)

LVQ and SOM are prototype-based competitive algorithms with supervised and unsupervised learning. Both algorithms typically compress any-dimensional data into a two-dimensional grid, however, only SOM takes taking into account data topology. LVQ employs labels in the training stage, the weights update can approximate or distance a prototype from an input pattern depending on if the categorization is correct or not [17]. SOM determines clusters based on the instances similarities and dissimilarities. Some previous studies suggest that supervised learning achieves superior performance than unsupervised learning for small training sets.

The proposed model is composed of multiple modules, each one consisting of a self-organizing map (SOM) trained for data clustering of a particular category, thus, there are as many SOM modules as the number of categories. Each SOM undergoes unsupervised training with samples belonging to a single category. Any node of each SOM module learns a prototype vector and a relevance vector associated with it. A second learning stage, the refinement phase, consists of mutual repelling of very close pairs of nodes belonging to different categories, closeness determined by distances below a threshold. Such distancing aims to increase the spatial separations of the different categories. After the two-stage training, each input pattern is presented to all SOM modules. The module with the winning node determines the category of the input instance.

After the two-stage learning process, each input pattern is presented to all SOM modules. The module with the winning node determines the category of the input instance.

3.1 The Description of SC-MuSOM

SC-MuSOM is formed by multiple modules of one-dimensional SOMs, each one defines a subcluster of a given category. For each SOM module, the weight vectors associated with each node are initialized as the arithmetic average of the instances of that category added to a random noise ranging within the interval $[-0.01 + 0.01]$ for each attribute. The relevance vector of each node is initialized as the complement of the variance vector (Eq. 1) of all instances of a considered

category. The calculation of the relevancies is an interactive process that takes into consideration this instantaneous term of relevance (a function of the variance) to update the current relevance of each dimension. The relevance value of each attribute weighs its distance component. Thus, dimensions with lower variance tend to be more relevant to determine the distance between an input vector and the subcluster centers (Eq. 2). Relevancies contribute to the calculation of distance so that certain attributes have greater importance without loss of information in a process of dimensionality reduction.

$$\mathbf{r}_k = (1 - \text{variance}(\mathbf{x}_k)) \tag{1}$$

$$d_r(\mathbf{x}_t, \mathbf{c}_k) = \sqrt{\sum \mathbf{r}_k (\mathbf{x}_{in} - \mathbf{c}_k)^2} \tag{2}$$

A SOM module groups any input pattern into a subcluster of a determined category. In any module, the winner node (w) is determined by the smallest Euclidean distance between an input pattern (\mathbf{x}_{in}) and each weight vector \mathbf{c}_k in cluster C (Eq. 3). In the training stage, the winner is searched among the nodes of the SOM module for each category. After the learning stage, any winner is the most active node of all categories. Algorithm 1 shows the first training stage.

$$\|\mathbf{x}_{in} - \mathbf{c}_w\| = \min_{k \in C} d_r(\mathbf{x}_{in}, \mathbf{c}_k) \tag{3}$$

The weight updating runs separately for each module, one by one, i.e., only the module of the input pattern category is updated. For all input vectors closest to c_w , a weight vector is adjusted with learning rate $\eta(t)$ and neighborhood function $h_w(t)$ as shown in Eqs. 4, 5, and 6. The excitation equation employs the relevance diagonal matrix \mathbf{R}_k , the relevancies form the main diagonal, in the update process.

$$\mathbf{c}_k(n + 1) = \mathbf{c}_k(n) + \eta(n) \cdot h_w(n) \cdot \mathbf{R}_k [\mathbf{x}_{in} - \mathbf{c}_k(n)] \tag{4}$$

$$\eta = \eta(0) \cdot e^{-\frac{n}{\tau_n}} \tag{5}$$

$$h_w(n) = h_0 \frac{e^{-\frac{\|\mathbf{c}_k - \mathbf{c}_w\|^2}{\sigma^2(n)}}}{\sigma^2(n)} \tag{6}$$

The neighborhood function, h_w decreases when the distance between c_k and c_w increases. We employed a Gaussian function to describe a topological neighborhood that shrinks with time due to the time varying dispersion term ($\sigma(n)$) defined by Eqs. 7.

$$\sigma(n) = \sigma_0 \cdot e^{-\frac{n}{\tau}} \tag{7}$$

For each epoch, all patterns are presented in random order, for each SOM module. At the end of each epoch, the values of all relevancies are updated. For relevance calculation, initially, all training patterns referring to each SOM are

evaluated and the node closest to each pattern is determined by Eq. 2. The new relevance (at time instant $n + 1$) associated with a node is calculated as a function of an instantaneous relevance (Eq. 1) that considers all patterns belonging to a subcluster. Each pair, weight vector and relevance vector, is associated with a subcluster, represented by a node. The weights are updated with each presentation of a new pattern whereas the relevancies are only updated at the end of each epoch.

The refinement procedure increases the distance between the closest prototypes of subcategories belonging to different categories. In this stage, the algorithm does not change the relevance vector since it is not affected by this slight movement. The relocation process runs for all nodes that have a distance below a threshold.

Algorithm 2 shows the refinement process, the second stage of learning. Initially, the cosines of the angles of all pairs of weight vectors belonging to different categories are calculated. Then, the pairs of nodes that are close enough and belong to different categories are selected for the separation process. The chosen distance metric is the cosine similarity metric [18]. Oppositely to the Euclidean distance, in the cosine similarity, the furthestmost vectors have a value nearby zero whereas the closest vectors have cosine neighboring 1. For a pair of vectors \mathbf{c}_k and \mathbf{c}_m , \mathbf{c}_k is moved away from \mathbf{c}_m (Eq. 8), movement determined by the minus sign in it. An analogous movement of \mathbf{c}_m with respect to \mathbf{c}_k is carried out swapping the roles of each vector in Eq. 8. The learning rate α is smaller than the rate η at the end of the first learning stage of learning. Equation 8 continues to use relevance in the weight refinement process, as in the Eq. 4.

$$\mathbf{c}_k(n + 1) = \mathbf{c}_k(n) + \alpha(n) \cdot \mathbf{R}_k [\mathbf{c}_m - \mathbf{c}_k(n)] \tag{8}$$

$$\alpha = \alpha(0) \cdot e^{-\frac{n}{\tau_\alpha}} \tag{9}$$

The categorization of an input vector is determined by the most active node in all SOM modules. This node determines the subcategory, therefore the category, to which the input belongs to. Clustering accuracy is calculated by the percentage of samples correctly categorized.

3.2 The Pseudo-code

The SC-MuSOM pseudo-code has two stages. The first stage, Algorithm 1, consists of module-by-module training. The updating of the relevance only occurs after all weights adjustment of an epoch.

Then, the pairs of nodes that have a cosine distance greater than a given threshold are selected to be pushed away. This process of calculating distance and moving away is repeated until the selected pairs are below the threshold (Algorithm 2).

Algorithm 1. SC-MuSOM-Stage1

```

1: Initialize all  $c_{kj}(0)$  with mean and noise insertion of  $[-0.01 +0.01]$  and learning
   parameters  $\eta(0)$ ,  $h_0$ ,  $\sigma_0$ ,  $\tau$ ,  $\tau_\eta$ ,  $T$ ;
2: for  $t = 0$  to  $T$  do
3:   for all  $x_t$  patterns from the category training set do
4:     Present pattern  $x_t$  at time  $t$ .
5:     Select the winning node for  $x_t$  by Eq. 3.
6:     Update the weights for all nodes by Eq. 4.
7:     Set  $t = t + 1$ ;
8:   end for
9:   Update the relevancies for all nodes by Eq. 1.
10:  Update the learning rate,  $\eta(t)$ , by Eq. 5.
11:  Update the topological neighborhood,  $\sigma(t)$ , by Eq. 7.
12: end for

```

Algorithm 2. SC-MuSOM-Stage2

```

1: Initialize  $\alpha(0)$  and  $Th$ ;
2: for  $t = 0$  to  $T$  do
3:   Calculate cosine distances from all nodes to all nodes.
4:   Selects pairs of nodes of different categories that have a cosine distance above
   a threshold  $Th$ .
5:   Updating the pair or the pairs selects pushing each other away by 8.
6:   Set  $t = t + 1$ ;
7: end for

```

4 Experiments

SC-MuSOM was tested with data mining and high-dimensional computer vision sets for performance assessment. We have run 10 rounds of tests with different random initializations for which we present the average performances with their standard deviations. We compared SC-MuSOM performance with those achieved by previously published articles. For a fair comparison, the number of executions for each algorithm follows that specified for published papers.

4.1 Datasets and Setups

The datasets Digits, Ionosphere, Iris, and Image Segmentation belong to the UCI repository [19]. The computer vision datasets include USPS [20], Scenes-15 [21] and Caltech-101 [22]. Table 1 shows the number of patterns (n_p), attributes (n_a), and categories (n_c) of each dataset and the type of each dataset.

The four data mining sets have low or medium dimensionality, ranging from 4 to 64 attributes. Such sets vary from 150 to 2,306 patterns divided into 2 to 10 categories. The patterns in the tree remaining sets have 256 to 1,080 attributes. Such high-dimensional data sets have 4,885 to 9,298 patterns that are categorized into 10 to 101 groups.

The digits data is composed of images. The Ionosphere set contains data on electrons in the ionosphere. The Iris dataset describes 3 different types of the plant Iris. The Image Segmentation dataset encodes several attributes of image regions. The USPS dataset stores images of handwritten digits. The Scene-15 dataset is formed by images for 15 outdoor scenes whereas the Caltech-101 dataset keeps images of objects belonging to 101 categories. For the Scenes-15 and Caltech-101 databases, features were extracted using a deep learning strategy through VGG-19 [24], a deep neural network. The output of the VGG-19 is a vector of 4,096 attributes.

Table 1. Datasets

Dataset	n_p	n_a	n_c	Type
Digits	1,797	64	10	Data mining
Ionosphere	351	34	2	Data mining
Iris	150	4	3	Data mining
Image segmentation	2,306	16	7	Data mining
USPS	9,298	256	10	Computer vision
Scenes-15	4,885	1,024	15	Computer vision
Caltech-101	8,677	1,080	101	Computer vision

4.2 Results

The performance of prototype-based models is shown in Table 2. SC-MuSOM performance is compared with the LGMLVQ [8] and KNN methods for low or medium-dimensional datasets. SC-MuSOM performance is also compared with those of KNN, deep-learning, and SOM-based soft subspace clustering methods (ETLMSC [23] and LARFDSSOM2 [4]) and the results are shown in Table 3. The tables bring mean accuracy values and their standard deviation. The numbers of the compared methods were collected in their original publications.

Table 2. Results for small and medium dimensional databases compared with KNN and LGMLVQ.

Dataset	KNN	LGMLVQ	SC-MuSOM (1 stage)	SC-MuSOM (2 stages)
Digits	94 (0.75)	87 (1.68)	97.09 (0.18)	97.09 (0.18)
Ionosphere	77 (4.68)	76 (3.75)	89.09 (4.12)	89.32 (5.67)
Iris	93 (3.40)	93 (2.98)	96.49 (2.56)	96.49 (2.53)
Image Seg.	93 (0.39)	94 (0.64)	90.43 (4.88)	91.90 (5.96)
USPS	94 (0.25)	95 (0.27)	94.07 (0.21)	94.17 (0.25)

Table 3. Results for high-dimensional databases compared with KNN, ETLMSC, and LARFDSSOM2.

Dataset	KNN	ETLMSC	LARFDSSOM2	SC-MuSOM (1 stage)	SC-MuSOM (2 stages)
Scenes-15	76.75 (1.08)	87.80 (0.2)	89.07 (0.2)	89.32 (0.15)	89.34 (0.14)
Caltech-101	60.07 (1.98)	63.9 (0.4)	65.41 (0.89)	74.47 (0.2)	74.50 (0.2)

4.3 Comparisons

We have not implemented the comparison methods shown in Table 2 and Table 3, therefore, the results in the tables are those reported in the literature.

SC-MuSOM reached the best performance for Digits, Ionosphere, Iris, Scenes-15 and Caltech-101 datasets. The performance of SC-MuSOM for USPS dataset was slightly below that of LGMLVQ. For the Image Segmentation dataset, SC-MuSOM achieved a performance inferior to KNN and LGMLVQ. The Image Segmentation dataset has a peculiar characteristic, its training data entails only 10% of the total set of patterns while the other sets have above 45% of the total of patterns used for training. The SOM compared to the LVQ tends to lose performance when the percentage of training data in relation to the total set of patterns is reduced [17].

An extension of the original LVQ model, LGMLVQ employs a complete relevance matrix in the similarity measure [8]. We consider a distance generalization that can explain the correlations between the attributes. SC-MuSOM only uses the calculation of feature vectors, being computationally less expensive in comparison with LGMLVQ.

The refinement stage of our algorithm marginally improves the performance of some of the tested databases. We apply the Mann-Whitney statistical test to verify whether the numerical differences are statistically significant. Such an analysis indicated that for 4 out of the 7 datasets, stage 2 presented statistically relevant improvements.

The promising results of the proposed method are due to the specialization in multi-modules of the SOM networks together with the use of relevances to make the distance metric discrimination more robust because of the curse of the dimensionality. The second stage of learning also contributes to a little performance improvement, this stage has a more significant impact on data mining bases than computer vision. The refinement stage is a low computational cost procedure that can marginally improve the results.

5 Discussion and Conclusion

Our proposed model, SC-MuSOM, has multiple SOM modules, one for each category of datasets used in the experiments. The first learning stage determines the clusters and their prototypes using the relevance vectors that assign different importance to the attributes. In the second stage, nodes of different categories

that are overlapped are moved away from each other. This second stage improves slightly the model accuracy.

SC-MuSOM is a soft subspace clustering (SSC) method that determines continuous values for relevancies ranging from zero to one. SC-MuSOM performs feature weighting to establish the relative prominence of each attribute regarding each category: the higher the relevance, the more important the attribute.

The refinement stage is characterized by the modification of the weight vectors by a low learning rate to separate further overlapping Voronoi regions between nodes of different modules. In some data mining and computer vision datasets, we notice a small improvement in accuracy compared to only the first stage of learning.

The model has a number of nodes established by the designer, however, the number of sub-categories may vary from category to category. This can make some nodes unnecessary. These nodes do increase the cost of running the algorithm. A potential improvement for this limitation is to employ a time-varying topology to increase the number of subcategories (nodes) for each category only if necessary. Something welcome to speed up the processing would be a massively parallel implementation of SC-MuSOM with GPUs or multi-core CPUs, another potential topic for future research.

References

1. Assent, I.: Clustering high dimensional data. *Wiley Interdisc. Rev. Data Min. Knowl. Discov.* **2**, 340–350 (2012)
2. Liu, C., Xie, J., Zhao, Q., Xie, Q., Liu, C.: Novel evolutionary multi-objective soft subspace clustering algorithm for credit risk assessment. *Expert Syst. Appl.* **138**, 112827 (2019)
3. Pereira, R.B., Plastino, A., Zadrozny, B., Merschmann, L.H.: Categorizing feature selection methods for multi-label classification. *Artif. Intell. Rev.* **49**, 57–78 (2018). <https://doi.org/10.1007/s10462-016-9516-4>
4. Araújo, A.F.R., Antonino, V.O., Ponce-Guevara, K.L.: Self-organizing subspace clustering for high-dimensional and multi-view data. *Neural Netw.* **130**, 253–268 (2020)
5. Deng, Z., Choi, K.S., Jiang, Y., Wang, J., Wang, S.: A survey on soft subspace clustering. *Inf. Sci.* **348**, 84–106 (2016)
6. Hammer, B., Villmann, T.: Generalized relevance learning vector quantization. *J. Int. Neural Netw. Soc.* **15**, 1059–68 (2002)
7. Melchert, F., Bani, G., Seiffert, U., Biehl, M.: Adaptive basis functions for prototype-based classification of functional data. *Neural Comput. Appl.* **32**, 18213–18223 (2020). <https://doi.org/10.1007/s00521-019-04299-2>
8. Schneider, P., Biehl, M., Hammer, B.: Adaptive relevance matrices in learning vector quantization. *Neural Comput.* **21**, 3532–3561 (2009)
9. Hammer, B., Schleif, F.M., Villmann, T.: On the generalization ability of prototype-based classifiers with local relevance determination. *Citeseer* (2005)
10. Bassani, H.F., Araújo, A.F.R.: Dimension selective self-organizing maps for clustering high dimensional data. In: *The International Joint Conference on Neural Networks*, pp. 1–8. IEEE (2012)

11. Bassani, H.F., Araújo, A.F.R.: Dimension selective self-organizing maps with time-varying structure for subspace and projected clustering. *IEEE Trans. Neural Netw. Learn. Syst.* **26**(3), 458–471 (2015)
12. Attaoui, M.O., Attaoui, M.O., Azzag, H., Lebbah, M., Keskes, N.: Subspace data stream clustering with global and local weighting models. *Neural Comput. Appl.* **33**(8), 3691–3712 (2021). <https://doi.org/10.1007/s00521-020-05184-z>
13. Araújo, A.F.R., Rego, R.L.: Self-organizing maps with a time-varying structure. *ACM Comput. Surv.* **46**(1), 7:1–7:38 (2013)
14. Hua, W., Lingfei, M.: Clustering ensemble model based on self-organizing map network. *Comput. Intell. Neurosci.* **2020** (2020)
15. Shahi, K.R., et al.: Hierarchical sparse subspace clustering (HESSC): an automatic approach for hyperspectral image analysis. *Remote Sens.* **12**(15), 2421 (2020)
16. Mishne, G., Talmon, R., Cohen, I., Coifman, R.R., Kluger, Y.: Data-driven tree transforms and metrics. *IEEE Trans. Sig. Inf. Process. Netw.* **4**(3), 451–466 (2017)
17. Goren-Bar, D., Kuflik, T.: Supporting user-subjective categorization with self-organizing maps and learning vector quantization. *J. Am. Soc. Inf. Sci. Technol.* **56**(4), 345–355 (2005)
18. Rahutomo, F., Kitasuka, T., Aritsugi, M.: Semantic cosine similarity. In: *International Student Conference on Advanced Science and Technology, ICAST*, vol. 4, no. 1 (2012)
19. Dua, D., Graff, C.: UCI machine learning repository (2017). <https://archive.ics.uci.edu/ml/index.php>
20. Hull, J.J.: A database for handwritten text recognition research. *IEEE Trans. Pattern Anal. Mach. Intell.* **16**(5), 550–554 (1994)
21. Fei-Fei, L., Perona, P.: A Bayesian hierarchical model for learning natural scene categories. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 524–531 (2005)
22. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In: *Conference on Computer Vision and Pattern Recognition Workshop*, p. 178 (2004)
23. Wu, J., Lin, Z., Zha, H.: Essential tensor learning for multi-view spectral clustering. *IEEE Trans. Image Process.* **28**(12), 5910–5922 (2019)
24. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: *International Conference on Learning Representations* (2015)