

Self-organizing subspace clustering for high-dimensional and multi-view data

Aluizio F.R. Araújo^{*}, Victor O. Antonino, Karina L. Ponce-Guevara^{*}

Centro de Informática, Universidade Federal de Pernambuco, 50740560, Recife, Brazil

ARTICLE INFO

Article history:

Received 10 December 2019
Received in revised form 30 April 2020
Accepted 28 June 2020
Available online 3 July 2020

Keywords:

Subspace clustering
Multi-view clustering
High-dimensional data
Self-organizing maps

ABSTRACT

A surge in the availability of data from multiple sources and modalities is correlated with advances in how to obtain, compress, store, transfer, and process large amounts of complex high-dimensional data. The clustering challenge increases with the growth of data dimensionality which decreases the discriminate power of the distance metrics. Subspace clustering aims to group data drawn from a union of subspaces. In such a way, there is a large number of state-of-the-art approaches and we divide them into families regarding the method used in the clustering. We introduce a soft subspace clustering algorithm, a Self-organizing Map (SOM) with a time-varying structure, to cluster data without any prior knowledge of the number of categories or of the neural network topology, both determined during the training process. The model also assigns proper relevancies (weights) to different dimensions, capturing from the learning process the influence of each dimension on uncovering clusters. We employ a number of real-world datasets to validate the model. This algorithm presents a competitive performance in a diverse range of contexts among them data mining, gene expression, multi-view, computer vision and text clustering problems which include high-dimensional data. Extensive experiments suggest that our method very often outperforms the state-of-the-art approaches in all types of problems considered.

© 2020 Elsevier Ltd. All rights reserved.

1. Introduction

High-dimensional datasets have been increasingly used in machine learning tasks with data that comes from images, videos, and text, among many sources. Such data often consist of a union of multiple low-dimensional structures rather than being uniformly distributed. The growth of dimensionality very often reduces the discriminant power of clustering distances (Liu et al., 2016); the dimension can be so large that the distance measurements can become meaningless, and therefore data can be miscategorized. Hence, clustering high-dimensional data may often discard a number of features in order to establish the groups. Thus, previous works considered reduction of the data dimensionality by creating new dimensions, combinations of the original attributes. This dimensionality reduction provokes loss of information, even while preserving information on irrelevant dimensions, thereby limiting the efficacy of such methods (Pereira et al., 2018).

Subspace clustering algorithms can be properly used for high-dimensional data space since they aim to identify clusters embedded in distinct subspaces (Kriegel et al., 2009). Subspace

clustering involves unsupervised learning for which a number of methods were applied to several fields such as data analysis, gene expression, multi-view data, and computer vision. These types of algorithms have employed a wide variety of machine learning techniques that can be grouped in categories or families (Deng et al., 2016; Kelkar & Rodd, 2019): Algebraic, Statistics, Attribute Weighted, Low-rank Representation (LRR), Spectral Algorithms (SSC), and Neural Networks (NN). Various algorithms for subspace clustering that will be compared with our model are briefly described in Section 2.2.

Particularly, some models based on neural networks have been employed for subspace clustering. They take advantage of a neural network's capacity for performing a relevant number of categorization tasks. Considering a timeline, the first algorithms are Generalized Relevance Learning Vector Quantization (GLVQ) (Hammer & Villmann, 2002) and Biclustering ARTMAP (BARTMAP) (Xu & Wunsch I.I., 2011). Then several SOM-based approaches were proposed: The Dimension Selective Self-organizing Map (DSSOM) (Bassani & Araújo, 2012) and the Local Adaptive Receptive Field Dimension Selective Self-organizing Map (LARFDS-SOM) (Bassani & Araújo, 2015). More recently, there are some approaches that use deep neural networks, such as Structured AutoEncoders for Subspace Clustering (StructAE) (Peng et al., 2018), Graph Structured Autoencoder (SSC) (Majumdar, 2018), and Semi-supervised Deep Embedded Clustering (SDEC) (Ren

^{*} Corresponding authors.

E-mail addresses: aluizioa@cin.ufpe.br (A.F.R. Araújo), klpg@cin.ufpe.br (K.L. Ponce-Guevara).

et al., 2019). There is also Saliency-aware Adaptive Resonance Theory (SA-ART), an ART-based model. In spite of the promising results obtained with the Neural Networks' models, there is room for improvement, especially concerning the applicability in different data sets, the training endeavor, and its use for sparse, high-dimensional, and multi-view data.

We introduce a soft subspace clustering algorithm, a Self-organizing Map with time-varying structure (Araújo & Rego, 2013), called the Local Adaptive Receptive Field Dimension Selective Self-Organizing Map 2 (LARFDSSOM2). It is an incremental learning method that is applicable for solving multiple types of subspace clustering problems. LARFDSSOM2 does not require prior knowledge of a problem, it is often faster for training than supervised methods like deep learning, and it handles high dimensional data and different levels of sparsity.

The original LARFDSSOM (Bassani & Araújo, 2015) was modified to improve its capacity to deal with high-dimensional and multi-view data, and to increase the algorithm applicability to different types of data. The main change includes a new way to determine dimension relevance that considers variance along each dimension for all data instances, instead of the probability distribution function of the attributes for each input pattern. Such a modification improves the capacity to consider intra and inter category relations between attributes. The learned variance throughout each dimension can simultaneously capture the similarities between dimension values in order to identify samples belonging to the same cluster and can determine differences between dimension values as discriminant factors to find distinct categories. LARFDSSOM2 also determines, without previous knowledge, the number of categories in which the space is divided. Furthermore, promising results are reported, for a number of datasets, significantly varying the number of data samples, the number of data features (attributes), and the number of expected groups.

The nodes of LARFDSSOM2 are dynamically added to or removed from the map during the algorithm learning stage. Then, a new cluster can be incrementally included or an existing node can be deleted according to the data being presented. LARFDSSOM2 can also self-adjust the receptive field of each node based on variance of the features, determining groups with distinct degrees of similarity and lengths of their basins of attraction. This establishes neighborhood relations for the groups based on the relevant dimensions for each category. Furthermore, LARFDSSOM2 assigns real-valued relevance weights to the attributes for identifying clusters as a soft subspace clustering algorithm does.

LARFDSSOM2 was tested in categorizing datasets of different domains and complexity. Also, it can provide insights into the relevance of each attribute, ranging between 0 and 1, determining discriminant features. The experiments included the generic datasets from the UCI machine learning repository, gene expression analysis, multi-view data, text clustering, and computer vision problems either grouping the images or segmenting them. LARFDSSOM2 was tested with binary and/or real attributes, and also for sparse datasets, a comprehensive range of tests considering previous research studies on analogous problems. It is worth mentioning that the same algorithm was used in various situations, demanding simply a change in the parameter settings.

In Section 2, the subspace clustering and multi-view clustering are defined and a number of their algorithms are briefly discussed. Section 3 explains how LARFDSSOM2 works. In Section 4, we present the experimental results and their analysis. Finally, Section 5 concludes the paper and makes some suggestions for future research.

2. Subspace clustering and multi-view clustering

Clustering aims to divide data into subsets (clusters) in which entities in the same subset are similar to each other while they are dissimilar to items in different groups. The level of similarity between data samples is often determined by distance metrics considering all attributes of the data samples (Kriegel et al., 2009). For high-dimensional data, such approaches disregard the fact that many dimensions can become irrelevant for clustering, misleading the clustering results since each sample is almost equidistant from the others. All-dimension proximity measures lose discriminant power with the growth of data dimensionality. Aggarwal et al. (2001) also argue that the relative distance from the farthest point and the nearest point tends to zero as consequence of dimension (d) growth. Hence, a cluster in that type of dataset can be determined according to a suitable subspace.

Other reasons for impairing clustering in high-dimensional space is that many attributes may become meaningless, e.g., some of them may be noise; a simple sample can be categorized in different groups, and a number of attributes can be correlated (Kelkar & Rodd, 2019). Such arguments reinforce the belief that distance measures used in conventional clustering algorithms can be replaced by adapted distances, suitable for subspace clustering.

Sometimes, scientific data analysis has to handle data obtained from various sources or from distinct feature extractors; often, they present heterogeneous properties. Consequently, such data samples can belong to different groups, depending on the features that determine a particular point of view. For instance, we can find different points of view for categorization of words in a document or of information describing documents (e.g. title, author, and sources). Thus, the learning process builds a function to model a particular view, which works in conjunction with other functions representing other views of the same dataset. Together, they aim to synergistically improve the learning performance in comparison with a single view dataset (Saha, 2013).

2.1. Definition, approaches, and algorithms for multi-view clustering

Multi-view clustering is defined as a data matrix $\chi = [(\chi^1)^T (\chi^2)^T \dots (\chi^t)^T] \in \mathbb{R}^{d \times n}$ with n data samples. This matrix contains t views sampled from c categories and $\chi^v = [\chi_1^v \chi_2^v \dots \chi_n^v] \in \mathbb{R}^{d_v \times n}$ denotes the v th view of χ with dimension d_v . The goal in multi-view clustering is to separate χ into c clusters by exploring the information in all t views of the input data.

A pioneer approach to deal with multi-view clustering is spectral clustering, featuring some algorithms which are worth mentioning. Shi and Malik (2000) introduced Normalized Cuts (Ncuts), a graph-based unsupervised learning algorithm in which points in the feature space are encoded by a weighted undirected graph. Two other spectral clustering methods are Co-regularized Multi-view Spectral Clustering (Kumar et al., 2011) and Co-training Approach for Multi-view Spectral Clustering (Kumar & Daumé, 2011). They use semi-supervised algorithms with co-regularization and co-training techniques. These models may not properly process missing data.

Several algorithms with linear complexity were proposed. Joint Non-negative Matrix Factorization (Liu et al., 2013) was applied to Multi-view Clustering in low-dimensional datasets. Wang et al. (2013) employed joint structured sparsity-inducing norms to determine the importance of each attribute considering different views. Yin et al. (2015) used Pairwise Sparse Subspace Representation with sparse datasets. Both models were effective for real-world datasets; however, they were not very competitive for sparse data.

Xie et al. (2018) introduced a method called Tensor-singular Value Decomposition based Multi-view Subspace Clustering (t-SVD-MSC). The authors determined a tensor by stacking the subspace representation matrices of different views and then rotating it, to capture the low-rank tensor subspace, so the high order correlations underlying multi-view data can be explored. This method was improved by Wu et al. (2019) who proposed a method named Essential Tensor Learning Method for Markov Chain based Multi-view Spectral Clustering (ETLMSC). The method constructs a transition matrix for each view to form a transition probability matrix tensor which is rotated and, then, the essential tensor is learnt.

Zhang et al. (2020) proposed a subspace clustering model to categorize multi-view data. Latent Multi-View Subspace Clustering (LMSC) takes advantage of the complementarity of multiple views according to two different utterances. Linear Latent Multi-view Subspace Clustering (LLMSC) adopts a linear relationship between the latent representation and the features from each view while Generalized Latent Multi-view Subspace Clustering (gLMSC) considers the nonlinear relationship captured by a neural network.

Zheng et al. (2020) proposed two models, Feature Concatenation Multi-view Subspace Clustering (FCMSC) and Graph Regularized FCMSC (gr-FCMSC); both build groups considering the multiple views simultaneously. The former explores the consensus information while the latter uses the complementary information of all views.

2.2. Subspace clustering: Definition, approaches, and algorithms

Vidal (2011) defined the subspace clustering problem as:

$$U_i = \{x_i \in \mathbb{R}^D : x_i = \phi_i + V_i y\}, \quad i = 1, \dots, n. \quad (1)$$

where $\{x_i \in \mathbb{R}^D\}_{i=1}^n$ is a collection of points; $\phi_i \in \mathbb{R}^D$ is a random point in subspace U_i , that can be set as $\phi_i = 0$ for linear subspaces; $V_i \in \mathbb{R}^D$ is the U_i basis; and $y \in \mathbb{R}^{d_i}$ is a low-dimensional representation of the data point x .

The algorithms can be divided into families, based on the method employed for clustering (Deng et al., 2016; Kelkar & Rodd, 2019): Algebraic-statistical, Attribute Weighted, Sparse Subspace Clustering, Low-rank Representation, Spectral Subspace Clustering, Robust Methods, and Neural Networks.

The Algebraic-statistical family aims to cluster data according to multiple subspaces using algebraic and geometric properties of a union of subspaces. The Attribute Weighting algorithms associate each cluster with a weighting vector to capture the subspace information of the cluster. Depending on the type of algorithm these weights may be fuzzy or based on entropy (Simiński, 2012). Another family is the Sparse Subspace clustering which is an extension of the Low-Rank Representation algorithms. The SSC and LRR methods constructed their affinity matrix by means of the representation coefficients in which the data can be structured by a linear combination of some data points of a common subspace. The main difference between SSC and LRR is the affinity matrix construction; the first one supposes that the matrix of representative coefficients is sparse while the second method supposes that the matrix of representative coefficients has low rank. One of the most popular algorithms is the Spectral Subspace Clustering. Such method divides the problem into two steps: learn an affinity data matrix and apply the spectral grouping to that affinity. The first step is considered the most important since the success of the spectral clustering algorithm depends on the construction of an informational affinity matrix (Elhamifar & Vidal, 2013). The family of Robust Algorithms aims to overcome various kinds of noises or other imperfections of the data. Such algorithms take advantage of the self-expressiveness property of linear subspaces

constructing a combination of the noiseless data points in each subspace. The Neural Networks family has algorithms developed with Adaptive Resonance Theory (ART) and Self-organizing Maps (SOMs).

One of the pioneer approaches for subspace clustering is the Algebraic-statistical family. We can initially mention the Random Sample Consensus (RANSAC) (Fischler & Bolles, 1981), a parameter estimator capable of pruning outliers in the input space for which the process complexity grows exponentially with dimensionality and a predefined number of subspaces. Sugaya and Kanatani (2004) introduced the Multi-Stage Learning algorithm (MSL) for multibody motion segmentation formed by a factorization method followed by subspace separation. MSL may get stuck in a local minimum. Vidal et al. (2005) presented an algebro-geometric algorithm for segmentation of subspaces with a time-varying number of dimensions. Generalized Principal Component Analysis (GPCA) is computationally cheap and copes with missing data; however, complexity grows exponentially with the number of subspaces and dimensions. The Agglomerative Lossy Compression (ALC) algorithm (Ma et al., 2007) assumes that data are drawn from a mixture of Gaussian distributions. ALC can handle noise and outliers in data and the number of subspaces and their dimensions are not known a priori. However, tuning its main parameter δ takes a significant computational effort.

The Attributes-Weighting Algorithm (AWA) (Chan et al., 2004), Fuzzy Weighting K-means (FWKM) (Jing et al., 2005), Entropy Weighting K-means (EWKM) (Jing et al., 2007), Locally Adaptive Metrics for Clustering (LAC) (Domeniconi et al., 2004), and Fuzzy Subspace Clustering (FSC) (Gan & Wu, 2008) are k-means type variable weighting methods capable of performing subspace clustering with an additional step in each iteration to calculate the weight values. AWA may suffer from zero dispersion on the feature, thus making the corresponding weight infinite. Furthermore, LAC, FWKM, and EWKM do not consider properties of shape volumes, not distinguished among clusters.

Soft subspace clustering algorithms have been proposed according to an attribute weighted approach. AFG-K-means (Gan & Ng, 2015) is an algorithm that calculates feature group weights and individual feature weights for clustering. It outperforms FG-K-means (Chen et al., 2012), incorporating automatic feature group selection. The model requires previous knowledge of the number of clusters. CKS-EWFC-K and CKS-EWFC-F (Wang et al., 2016) have a mechanism for distance metric learning. The distance between samples with respect to each feature can be learned by using a linear combination of a set of basis kernel mappings. The results were promising, however the model was not tested with sparse data.

Subspace Weighting Co-Clustering (SWCC) (Liu et al., 2018a), was applied to high dimensional gene expression data. In SWCC, a gene subspace weight matrix is introduced to identify the contribution of gene objects (attributes) in distinguishing different sample clusters. SWCC assigns weights to the gene objects on each sample cluster by using a weight matrix capable of identifying each individual gene. They use subspace weights in the distance function to determine the co-clusters of data and optimize the process by automatically computing the subspace weights.

Among the sparse methods, we can comment on Sparse Subspace Clustering (SSC) (Elhamifar & Vidal, 2009), Local Subspace Affinity (LSA) (Yan & Pollefeys, 2006), and Spectral Curvature Clustering (SCC) (Chen & Lerman, 2009). They define any particular point as a linear combination of neighbor points. SCC and LSA need to know the number of subspaces in advance, while SSC may yield a point which is noisy and contaminated by outliers as a combination of other points (Vidal & Favaro, 2014).

Low-rank Representation Methods form another family of methods. A Unified Framework for Representation-Based Subspace Clustering presents three methods (Peng et al., 2015): (1) scalable SSC (SSSC), (2) scalable LRR (SLRR), and (3) scalable least square regression (LSLR). They treat a large-scale problem as an out-of-sample problem since they do sampling and clustering. However, the methods can fail when that out-of-sample data comes from a new subspace. A correntropy-based structured LRR (CLRR) (He et al., 2015), introduced information theoretic objective functions in order to combine structured low-rank representations (LRRs) capturing the global structure of data. The authors claimed they can handle outliers.

Low Rank Subspace Clustering Method via Discrete Constraint and Hypergraph Regularization, DHLRS (Liu et al., 2018b), is a recent approach that does a subspace clustering using a low-rank representation approach. Its experiments aim to determine tumor molecular pattern. It obtains the data low-rank representation via discrete constraint, learning the cluster indicators instead of the affinity matrix.

Talking about robust approaches, we mention Robust Subspace Clustering by Cauchy Loss Function (RCLF) (Li et al., 2018) that aims to minimize the noise found in the nature of real data by implementing a subspace clustering based on the Cauchy loss function (CLF). In spite of handling noisy data, RCLF does not find some subspace structures in the data.

There are also the algorithms developed with deep neural networks. StructAE (Peng et al., 2018) trains a neural network to project the input into another subspace, clustering the data into multiple subspaces. Graph Structured Autoencoder (SSC) (Majumdar, 2018) is a graph regularized version of autoencoder with several variants of the basic one. Semi-supervised Deep Embedded clustering (SDEC) (Ren et al., 2019) employs pairwise constraints to impose increasing the of proximity of samples in the same cluster and remoteness of data samples from different clusters. This kind of algorithm shows promising performance for image clustering.

For the neural network family, we can mention Saliency-aware Adaptive Resonance Theory (SA-ART) (Meng et al., 2019) employed for clustering large-scale sparse data. SA-ART takes the Fuzzy ART and adds three statistical measures to its functions of similarity measure and cluster weight modeling in order to identify cluster-wise salient features. The authors employed three simple statistical measures, frequency, mean, and variance, to estimate the importance and distributions of features for each cluster.

Other neural network-based methods include Self-organizing Maps. More specifically, Dimension Selective Self-organizing Map (DSSOM) (Bassani & Araújo, 2012) and Local Adaptive Receptive Field Dimension Selective Self-organizing Map (LARFDSSOM) (Bassani & Araújo, 2015). Both are characterized by the relevance of each dimension that is learned during the training stage. DSSOM demands predetermination of the number of clusters; nevertheless, this is an emerging property of the LARFDSSOM. Both models were not previously tested for sparse data.

We briefly presented instances of the two main classes of subspace clustering algorithms: the hard and the soft ones. Deng et al. (2016) surveyed the literature on soft subspace clustering (SSC), a current tendency to handle clustering in high-dimensional spaces. Soft subspace clustering assigns weights to each dimension of the considered space, allowing different and continuous contributions of each dimension of a cluster.

LARFDSSOM is a type of independent SSC (Deng et al., 2016) in which each cluster has an independent subspace. LARFDSSOM is easy to implement and produced promising results for a number of data mining sets (Bassani & Araújo, 2015). This algorithm is also employed in image clustering problems (Medeiros

et al., 2019). The authors proposed an unsupervised visual object categorization (UVOC) method that aims to correctly separate different categories. In this context, LARFDSSOM is used for clustering disjoint and non-disjoint images. Furthermore, Bassani and Araújo (2019) employed LARFDSSOM for Natural Language Processing (NLP) to generate codebook vectors and, then, group the images according to a given label. In the present article we aim to modify LARFDSSOM to widen the range of use for the approach, applying it to more complex sets (increasing dimensionality combined with a growth of the number of clusters and/or number of samples), different domains (data mining, gene expression, multi-view data, text clustering, and computer vision), different samplings (sets with few and many samples), and distinct density (dense and sparse data).

Table 1 shows a summary of our review of methods for subspace clustering, their families and the type of problems for which they were applied.

3. LARFDSSOM2

A Self-Organizing Map (SOM) (Kohonen, 1990) is a clustering algorithm to reach a dimensional reduction in the output space following the data distribution of the input space. SOM needs predefinition of the topology and maximum number of clusters, two limiting factors for mapping accuracy (Araújo & Rego, 2013) since such predefinitions demand detailed knowledge of the input space. Hence, some researchers have proposed SOMs with growing structure.

We define LARFDSSOM2 based on a framework to describe Self-Organizing Maps with time-varying structures (Araújo & Rego, 2013). Its key difference from the original LARFDSSOM (Bassani & Araújo, 2015) is the calculation of the relevance of the dimensions considering variance throughout the dimensions instead of considering the probability distribution function of each input pattern.

LARFDSSOM2 aims to cluster samples presented to it. Such samples in a considered dataset can belong to more than one cluster. LARFDSSOM2 creates clusters considering the relevance of each dimension, so, each cluster is determined based on its own subset of relevant features which are learned by the network. Hence, LARFDSSOM2 discovers the number of clusters, learns the level of relevance of each dimension for each cluster, and identifies the groups according to such relevancies. It is worth mentioning that the dimensionality of the samples (and of the problem) is not reduced; instead, for each cluster, the features are weighted, so unimportant dimensions for determining a cluster become irrelevant to calculate the distance between a sample and the current set of cluster prototypes.

The main difference between LARFDSSOM2 and its predecessor is indeed the way the relevancies are calculated. LARFDSSOM had a significant loss of performance for sparse data because it loses discriminant power for this case. On the other hand, the variance throughout each dimension can capture the similarities between dimension values (important for identification of samples belonging to a same cluster) and can identify differences between dimension values (relevant for distinguishing different categories). The calculated dimension relevancies consider both factors.

LARFDSSOM2 is composed of three stages: organization, convergence, and clustering. The flowchart sketches the two main steps of the algorithm. The training stages of LARFDSSOM2 are detailed by Algorithm 1 and Algorithm 2. The procedure for clustering a dataset is detailed by Algorithm 3.

The organization stage (Algorithm 1) aims to determine the cluster of each input signal, to adjust the relevance and prototype vectors of each cluster, and to remove clusters (nodes in the map)

Table 1

Type of application, Subspace clustering family, and methods. The contexts are represented as SV, MV, CV, GE, TC, meaning single-view, multi-view, computer vision, gene expression and text clustering, respectively.

Context	Family	Model
SV	Attribute weighted	AWA (Chan et al., 2004), CKS-EWFC-K (Wang et al., 2016), CKS-EWFC-F (Wang et al., 2016), LAC (Domeniconi et al., 2004), FWKM (Jing et al., 2005), EWKM (Jing et al., 2007), FSC (Gan & Wu, 2008).
	Neural networks	LARFDSSOM (Bassani & Araújo, 2015).
MV	Algebraic-statistical	Multi NMF (Liu et al., 2013)
	Low-rank representation	t-SVD-MS (Xie et al., 2018), ETLMSC (Wu et al., 2019), ILMSC and gLMSC (Zhang et al., 2020), FCMSC and gr-FCMSC (Zheng et al., 2020).
	Spectral SSC	S Spectral, S Subspace (Shi & Malik, 2000), Co-Training (Kumar & Daumé, 2011)
	Sparse methods	MultiSSC (Yin et al., 2015), Multi CF (Wang et al., 2013), PairwiseSC, CentroidSC (Kumar et al., 2011)
	Neural networks	LARFDSSOM (Bassani & Araújo, 2015).
GE	Attribute weighted	AWA (Chan et al., 2004), LAC (Domeniconi et al., 2004), FWKM (Jing et al., 2005), EWKM (Jing et al., 2007), FSC (Gan & Wu, 2008), FG-K (Chen et al., 2012), AFG-K (Gan & Ng, 2015), SWCC (Liu et al., 2018a)
	Low-rank representation	DHLRS (Liu et al., 2018b)
CV	Algebraic-statistical	RANSAC (Fischler & Bolles, 1981), MSL (Sugaya & Kanatani, 2004), GPCA (Vidal et al., 2005), ALC (Ma et al., 2007)
	Low-rank representation	SSSC, SLRR, SLRS (Peng et al., 2015), t-SVD-MS (Xie et al., 2018), ETLMSC (Wu et al., 2019), ILMSC and gLMSC (Zhang et al., 2020), FCMSC and gr-FCMSC (Zheng et al., 2020).
	Sparse methods	LSA (Yan & Pollefeys, 2006), SSC (Elhamifar & Vidal, 2009)
	Neural networks	StructAE (Peng et al., 2018), SSC (Majumdar, 2018), SDEC (Ren et al., 2019), SA-ART (Meng et al., 2019).
TC	Low-rank representation	SSSC, SLRR, SLRS (Peng et al., 2015), t-SVD-MS (Xie et al., 2018), ILMSC and gLMSC (Zhang et al., 2020), FCMSC and gr-FCMSC (Zheng et al., 2020)
	Neural networks	SA-ART (Meng et al., 2019)

that seldom fire. Initially, the algorithm parameters and vectors of variables are initialized (lines 1 to 3). The initialized parameters are reported in line 1: (i) activation threshold, a_t , that determines the minimum similarity between a prototype and the members of its cluster; (ii) the learning rates of the winner node, e_b , and its neighbors, e_n , to adapt relevance and prototype vectors; (iii) the parameter β , determining the change rate of a moving average to calculate the relevancies; (iv) the slope, s_l , of the relevance vector equation, a sigmoidal expression; (v) the maximum number of nodes (number of clusters) in the map, N_{max} ; (vi) the maximum number of competitions, $maxcomp$, a node can win; (vii) the percentage lp determining the lowest number of wins that a node needs to remain in the map without being pruned; and (viii) the number of samples, t_{max} , to be presented to the map. Line 2 shows the initialization of the vectors of variables associated with the only node that starts the map: prototype or center vector; the distance vector; the relevance vector; the maximum, minimum, and mean relevance vectors; and the number of competitions since the last map reset and the current number of nodes in the map.

Lines 5 to 12 describe the determination of the node with highest activation (a_s), the winner node of a competition to respond to an input sample. Then, if $a_t < a_s$, the algorithm inserts a new node with its initialization vectors to represent a new cluster; otherwise, the vectors associated with the winner node and its neighbors are updated. Finally, lines 13 to 16 show the removal of nodes that won few competitions and the necessary re-initializations. The parameter values for Algorithms 1, 2, and 3 are the same.

In the convergence stage (Algorithm 2), the nodes with few victories are removed from the map; the connections between the remaining nodes are updated and the network is retrained without inserting any new node. In the clustering stage (Algorithm 3), each sample is presented to the network and the winners determine the categories to which the sample belongs.

LARFDSSOM2 follows pseudo-code that has the same sequence of steps of LARFDSSOM (Bassani & Araújo, 2015). The main details and the modifications are described in the following subsections.

Fig. 1 sketches the flowchart for LARFDSSOM2 which includes the two phases of the algorithm: self-organization and convergence, Algorithms 1 and 2 respectively.

Algorithm 1: SELF-ORGANIZATION PHASE

```

1 Assign values to parameters  $a_t$ ,  $e_b$ ,  $e_n$ ,  $\beta$ ,  $s_l$ ,  $N_{max}$ ,  $maxcomp$ ,  $lp$ ,  $t_{max}$ ;
2 Insert a node in the map:  $\mathbf{c}_j \leftarrow \mathbf{x}$  and  $\delta_j \leftarrow \mathbf{0}$ ,  $\omega_j \leftarrow \mathbf{1}$ ,  $\phi_{jmax} \leftarrow \mathbf{0}$ ,  $\phi_{jmin} \leftarrow \mathbf{0}$ ,  $\phi_{jmean} \leftarrow \mathbf{0}$  and  $wins_j \leftarrow 0$ ;
3 Give initial values to  $nwins \leftarrow 1$ ,  $N \leftarrow 0$ ;
4 for  $t \leftarrow 0$  to  $t_{max}$  do
5   Draw a sample  $\mathbf{x}$  and show it to the map;
6   Calculate the activation function for each node;
7   Determine the highest active node,  $s$ , for which  $a_c = a_s$ ;
8   if  $a_s < a_t$  and  $N < N_{max}$  then
9     Insert a node  $j$  defined as  $\mathbf{c}_j \leftarrow \mathbf{x}$ ,  $\delta_j \leftarrow \mathbf{0}$ ,  $\omega_j \leftarrow \mathbf{1}$ ,  $\phi_{jmax} \leftarrow \mathbf{0}$ ,  $\phi_{jmin} \leftarrow \mathbf{0}$ ,  $\phi_{jmean} \leftarrow \mathbf{0}$  and  $wins_j \leftarrow lp \times nwins$ ;
10    Create links of node  $j$  its neighbors  $N \leftarrow N + 1$ 
11  end
12  else
13    Adjust the vectors associated with winner and neighbors:  $\delta_s$ ,  $\phi_{smax}$ ,  $\phi_{smin}$ ,  $\phi_{smean}$ ,  $\omega_s$ , and  $\mathbf{c}_s$ ;
14  end
15  if  $nwins = maxcomp$  then
16    Delete all nodes for which  $wins_j < lp \times maxcomp$ ;
17    Remove connections of the seldom activated nodes;
18    Assign zero to the number of wins of all other nodes:  $wins_j \leftarrow 0$ ;
19     $nwins \leftarrow 0$ 
20  end
21   $nwins \leftarrow 0$ ;
22 end

```


Algorithm 2: CONVERGENCE PHASE

```

1 while true do
2    $N_{max} \leftarrow N$ ; Delete all nodes for which
    $wins_j < lp \times maxcomp$ ;
3   if ( $N = N_{max}$  or  $N = 1$ ) return;
4   Modify connections between nodes to update topology;
5   Assign zero to the number of wins of all other nodes:
    $wins_j < lp \leftarrow 0$ ;
6   for  $t \leftarrow 0$  to  $t_{max}$  do
7     Draw a sample  $\mathbf{x}$ , show it to the map and calculate
     all activations;
8     Determine the winner node  $s$ ;
9     Adjust the vectors associated with winner and
     neighbors:  $\delta_s, \phi_{smax}, \phi_{smin}, \phi_{smean}, \omega_s$ , and  $\mathbf{c}_s$ ; Set
      $wins_s \leftarrow wins_s + 1$ ;
10  end
11 end

```

Algorithm 3: CLUSTERING WITH LARFDSSOM2

```

1 foreach sample  $\mathbf{x}$  do
2   Show  $\mathbf{x}$  to the map; Calculate the activation functions;
3   Determine the winner  $s$ ;
4   if  $a_s \geq a_t$  then
5     repeat
6       Include  $\mathbf{x}$  to the  $s$ -th cluster;
7       Determine the second most active node, the
       second winner;
8     until  $a_s < a_t$ ;
9   end
10  Include  $\mathbf{x}$  in the group of outliers;
11 end

```

The sample $\mathbf{x}$ belongs to the cluster j if the value a_s is greater than a_t . If this is true, the weight vector adaptation procedure is triggered; otherwise, the growing procedure is triggered to insert a new node.

3.3. Weight vectors adaptation procedure

After determining the winner node, the vectors associated to it and its neighbors are updated. Initially, the weight vectors are:

$$\mathbf{c}_j(n+1) = \mathbf{c}_j(n) + e(\mathbf{x} - \mathbf{c}_j(n)) \quad (5)$$

where $e = e_b$ or $e = e_n$, $e \in [0, 1]$ are learning rates for the winner and its neighbors. The distance vectors are adjusted considering

a moving average term:

$$\delta_j(n+1) = (1 - e\beta)\delta_j(n) + e\beta(|\mathbf{x} - \mathbf{c}_j(n)|) \quad (6)$$

where e is the learning rate, either e_b or e_n , $\beta \in [0, 1]$ modulates the moving average, and $|\cdot|$ is the magnitude vector of $(\mathbf{x} - \mathbf{c}_j(n))$.

The intraclass correlation is often used to quantify the level to which patterns with a fixed degree of relatedness resemble each other in terms of a quantitative trait. Hence, variance is a candidate measure to determine the relevance of each dimension. Parsons et al. (2004) argue that the features that are not important to determine the clusters have high variance values within each considered cluster while such attribute values are uniformly distributed on the whole data space. Taking into consideration a particular cluster, the values of dispersions for relevant features are compactly distributed; on the other hand, they become scattered for irrelevant attributes. In the context of the intraclass correlation for LARFDSSOM2, an attribute is relevant to a particular cluster when its variance in the components of the distance vector is small. For the interclass correlation, this analysis would be reversed.

After updating the distance vectors, each component of the relevance vector (ω_{ij}) is calculated by a logistic function of the distances δ_{ji} , an approximation of the median absolute deviation (MAD). This statistical dispersion estimate is more robust to outliers than the standard deviation because in the latter the distances from the mean are squared, emphasizing the distortion caused by outliers. This is the main reason that Bassani and Araújo (2015) employed Eq. (7) to capture the attribute relevancies:

$$\omega_{ji} = \begin{cases} \frac{1}{1 + \exp\left(\frac{\delta_{jmean} - \delta_{ji}}{s_l(\delta_{jmax} - \delta_{jmin})}\right)}, & \text{if } \delta_{jmin} \neq \delta_{jmax} \\ 1, & \text{otherwise} \end{cases} \quad (7)$$

where δ_{jmax} , δ_{jmin} and δ_{jmean} are the maximum, minimum, and average values of each element of a distance vector. A relevance component, ω_{ij} , varies between zero and one and $s_l > 0$ determines the steepness of the curve.

In spite of their suitable properties, Eq. (7) is asymmetric and directional; therefore, it yields different results for an input with inverted signal and the same discrepancy module. Moreover, Eq. (7) tends to generate binary relevancies, values close to 0 or 1. Hence, we dropped Eq. (7) and we calculate relevancies using a radial basis function in which the outputs vary monotonically with distances from a central point:

$$\omega_{ji} = \begin{cases} \frac{1}{\exp\left(\frac{(\phi_{jmean} - \delta_{ji})^2}{s_l(\phi_{jmax} - \phi_{jmin})}\right)}, & \text{if } \phi_{jmin} \neq \phi_{jmax} \\ 1, & \text{otherwise} \end{cases} \quad (8)$$

All three vectors ϕ_{jmax} , ϕ_{jmin} and ϕ_{jmean} are initialized with zero, and they are updated when the node j undergoes a change in its distance vector. Each ϕ_{jmax} component is updated by making a comparison between its previous value and the current δ_{ji} . The highest value between the two is assigned to ϕ_{jmax} . An analogous procedure is considered to calculate ϕ_{jmin} while ϕ_{jmean} is determined by adding all previous distance vectors and dividing the summation by the number of updates that the j node underwent:

$$\phi_{jmax} = \begin{cases} \delta_{ji}, & \text{if } \delta_{ji} > \phi_{jmax} \\ \phi_{jmax}, & \text{otherwise} \end{cases} \quad (9)$$

$$\phi_{jmin} = \begin{cases} \delta_{ji}, & \text{if } \delta_{ji} < \phi_{jmin} \\ \phi_{jmin}, & \text{otherwise} \end{cases} \quad (10)$$

$$\phi_{jmean} = \frac{\sum_{t=1}^{n_{act}} \delta_{jmean}^t}{n_{act}} \quad (11)$$

Eqs. (9) to (11) determine an accurate measure of variability that is robust to outliers. The relevancies range continuously from zero to one.

3.4. Topology updating procedure

The connections between nodes determine the neighborhoods, established by the closeness of the relevance vectors of each node:

$$i \text{ and } j \text{ are } = \begin{cases} \text{connected,} & \text{if } \|\omega_i - \omega_j\| < c\sqrt{m} \\ \text{disconnected,} & \text{otherwise} \end{cases} \quad (12)$$

where m is the dimension of the input space and c is the threshold for the links.

The competition and cooperation steps are repeated for a limited number of epochs, and during this process, the nodes that do not win a minimum number of patterns are periodically removed from the map.

4. Experiments

In this paper, we propose LARFDSSOM2 for subspace clustering in high-dimensional and multi-view data. The proposed algorithm was tested with a number of datasets which include applications of data mining, gene expression, multi-view categorization, and computer vision. Information on the features of the datasets are shown in Table 2. All programs can be found at <https://github.com/klponce/larfdssom2.git>

In order to evaluate the statistical significance of the experimental results, rounds of tests with different random initializations were conducted and the average performances with their standard deviations are shown. It is worth stressing that we compared LARFDSSOM2 with results obtained from previously published articles. Hence, the number of LARFDSSOM2 executions for each dataset is the same as previously informed by the literature for each comparison case. Hence, considering the experiments chosen for comparison, LARFDSSOM2 was run 10 times when dealing with UCI datasets (Dua & Graff, 2017), 20 times with gene expression datasets, and 100 times with multi-view and computer vision datasets.

The remainder of this section starts with Section 4.1 that defines the clustering measures used in the experiments. Sections 4.2–4.4 describe data mining experiments using single-view, gene expression, and multi-view datasets. Section 4.5 concerns the results on computer vision data. Finally, Section 4.7 discusses parametric adjustments. From Tables 3 to 10, bold members indicate best results and values in parentheses are standard deviations.

4.1. Quality measures

The categorization quality of the different clustering methods is measured using one of the following metrics: Normalized Mutual Information (NMI), Clustering Accuracy (ACC), Clustering Error (ERR), Rand Index (RI), and Adjusted Rand Index (ARI). The results presented in the next subsections employ at least one of such metrics to produce the comparisons. As we only ran all experiments for LARFDSSOM and LARFDSSOM2, we used the results published for the previous studies for comparisons. Therefore, the choices of the metrics are the same as used in the original experiments (previous papers) of the algorithms compared with LARFDSSOM2 for each dataset. In detail, the used performances were the RI (Table 3), ARI (Table 4), ACC (Table 6), NMI (Table 7), and ERR (from Tables 9 to 11). Furthermore, for the computer vision experiments, the category labels are available, so the clustering accuracy was measured in ACC, NMI, and ARI.

Table 2

Datasets description. The n_s ; n_a ; n_c ; t_p refer to the number of samples, attributes, clusters, and the type of problem, respectively.

Dataset	n_s	n_a	n_c	t_p
Australian	690	15	2	Single-view
Breast tissue	106	9	6	Single-view
Bupa	345	6	2	Single-view
Heart	270	13	2	Single-view
Iris	150	4	3	Single-view
Parkinsons	195	23	2	Single-view
Pima Indians diabetes	768	8	2	Single-view
Vehicle	846	18	4	Single-view
Wdbc	569	30	2	Single-view
Wine	178	13	3	Single-view
Letter-ABCD	3,096	16	4	Single-view
Letter-UVWX	3,116	16	4	Single-view
Alizadeh-2000-v2	62	2,093	3	Gene expression
Gordon-2002	181	1,626	2	Gene expression
Armstrong-2002-v1	72	1,081	2	Gene expression
Chowdary-2006	104	182	2	Gene expression
Leukemia	72	7,129	2	Gene expression
Prostate Cancer	102	6,033	2	Gene expression
Brain_Tumor2	50	10,367	4	Gene expression
Digits	2,000	292	10	Multi-view
Movies617	617	3,276	17	Multi-view
WebKB	187	2,264	5	Multi-view
Animal	6,089	4,252	10	Multi-view
BBCSport	737	4,613	5	Multi-view
Caltech-101	8,677	1,080	101	Multi-view
MITIndoor-67	15,620	1,024	67	Multi-view
Scene-15	4,885	1,024	15	Multi-view
Extended Yale B faces	2,414	2,016	38	Computer vision
CIFAR 10	60,000	3,072	10	Computer vision
COIL 20	1,440	256	20	Computer vision
Imagenet	64,000	74,506	50	Computer vision
BlogCatalog	66,418	17,824	147	Text clustering

4.2. Single-view data mining data experiments

The performance of LARFDSSOM2 is compared with eight existing clustering algorithms using twelve single-view datasets from the UCI repository. The algorithms for comparisons are LAC (Domeniconi et al., 2004), EWKM (Jing et al., 2007), FSC (Gan & Wu, 2008), AWA (Chan et al., 2004), FWKM (Jing et al., 2005), and both CKS-EWFC-K, CKS-EWFC-F (Wang et al., 2016) and LARFDSSOM. The samples of the datasets were divided into 75% for training and 25% for tests. LARFDSSOM2 has shown comparable or better results (9 out 12 times) than all other algorithms for all considered datasets (Table 3). LARFDSSOM2 shows better performance than LARFDSSOM for all cases.

4.3. Gene expression experiments

To categorize gene expression data, machine learning algorithms aim to find categories of samples sharing similar expression patterns. LARFDSSOM2 was tested with four datasets: Alizadeh-2000-v2 (Alizadeh et al., 2000), Gordon-2002 (Gordon et al., 2002), Armstrong-2002-v1 (Armstrong et al., 2002), and Chowdary-2006 (Chowdary et al., 2006). They are characterized by few input samples, two or three categories, and the number of attributes ranging from 192 to 2093. This is a scenario quite different from that in Section 4.2. The clustering methods chosen for comparison are LAC (Domeniconi et al., 2004), FSC (Gan & Wu, 2008), EWKM (Jing et al., 2007), FG-k-means (Chen et al.,

Table 3

Summary of clustering results (RI) on twelve datasets by nine algorithms (CKS-EWFC-K; CKS-EWFC-F; EWKM; FWKM; AWA; FSC; LAC; LARFDSSOM, and LARFDSSOM2). Numbers in parenthesis are the standard deviations. Bold numbers indicate the best results.

RI (%)	CKS-K	CKS-F	EWKM	FWKM	AWA	FSC	LAC	LF	LARFDSSOM	LARFDSSOM2
Australian	0.71(0.07)	0.72(0.08)	0.59(0.03)	0.54(0.0)	0.55(0.19)	0.50(0.0)	0.52(0.01)	0.56(0.1)	0.55(0.03)	0.76 (0.01)
Breast tissue	0.81(0.02)	0.74(0.02)	0.72(0.02)	0.60(0.04)	0.76(0.03)	0.73(0.02)	0.71(0.02)	0.78(0.2)	0.64(0.02)	0.91 (0.00)
Bupa	0.51(0.0)	0.50(0.0)	0.51(0.0)	0.49(0.0)	0.50(0.0)	0.50(0.0)	0.51(0.0)	0.48(0.0)	0.46(0.05)	0.54 (0.0)
Heart	0.69(0.0)	0.68(0.0)	0.65(0.0)	0.57(0.0)	0.56(0.0)	0.54(0.0)	0.67(0.0)	0.58(0.0)	0.63(0.06)	0.75 (0.0)
Iris	0.92(0.0)	0.87(0.0)	0.86(0.0)	0.90(0.0)	0.94 (0.0)	0.93(0.0)	0.86(0.0)	0.83(0.0)	0.85(0.02)	0.90(0.0)
Parkinsons	1.0 (0.0)	0.97(0.0)	0.66(0.11)	0.56(0.0)	1.0 (0.0)	0.62(0.0)	0.61(0.0)	0.91(0.0)	0.89(0.01)	0.95(0.0)
Diabetes	0.61(0.0)	0.55(0.0)	0.55(0.0)	0.53(0.0)	0.53(0.0)	0.53(0.0)	0.54(0.0)	0.53(0.0)	0.54(0.04)	0.64 (0.0)
Vehicle	0.66(0.0)	0.66(0.0)	0.65(0.0)	0.64(0.0)	0.67(0.0)	0.66(0.0)	0.64(0.0)	0.55(0.0)	0.63(0.01)	0.68 (0.0)
wdbc	0.89 (0.0)	0.86(0.0)	0.83(0.0)	0.83(0.0)	0.79(0.0)	0.75(0.1)	0.84(0.0)	0.74(0.0)	0.74(0.02)	0.88(0.0)
Wine	0.94 (0.0)	0.89(0.0)	0.68(0.0)	0.71(0.0)	0.85(0.0)	0.80(0.0)	0.83(0.0)	0.88(0.0)	0.83(0.02)	0.92(0.0)
Letter-ABCD	0.79(0.0)	0.76(0.0)	0.77(0.0)	0.75(0.0)	0.71(0.0)	0.72(0.0)	0.77(0.0)	0.65(0.0)	0.81(0.01)	0.86 (0.0)
Letter-UVWX	0.71(0.0)	0.70(0.0)	0.70(0.0)	0.70(0.0)	0.64(0.0)	0.65(0.0)	0.70(0.0)	0.71(0.0)	0.63(0.04)	0.76 (0.0)

Table 4

Summary of clustering results (ARI) on four datasets by seven algorithms. Numbers in parenthesis are the standard deviations. Bold numbers indicate the best results.

ARI (%)	Alizadeh-2000-v2	Gordon-2002	Armstrong-2002-v1	Chowdary-2006
FSC	0.5 (0.27)	0.5 (0.42)	0.18 (0.23)	0.54 (0.32)
LAC	0.59 (0.19)	0.57 (0.48)	0.32 (0.28)	0.06 (0.02)
EWKM	0.59 (0.2)	0.53(0.46)	0.32(0.28)	0.06(0.02)
FG-k-means	0.56(0.19)	0.54(0.46)	0.28 (0.25)	0.06 (0.02)
AFG-k-means	0.54 (0.22)	0.69 (0.39)	0.33 (0.32)	0.76 (0.24)
LARFDSSOM	0.66 (0.23)	0.62 (0.19)	0.27 (0.25)	0.57 (0.27)
LARFDSSOM2	0.85 (0.36)	0.82 (0.76)	0.65 (0.22)	0.96 (0.34)

Table 5

Summary of clustering results (ACC) on three datasets by four algorithms. Numbers in parenthesis are the standard deviations. Bold numbers indicate the best results.

ACC %	Leukemia	Prostate	BrainTumor2
DHLRS	79.39 (0.00)	63.73 (0.00)	58.00 (3.53)
SWCC	67.00(0.01)	51.00(0.54)	73.00(0.86)
LARFDSSOM	82.12 (1.21)	58.92 (1.01)	63.92 (1.16)
LARFDSSOM2	85.63 (0.62)	66.51 (1.22)	79.22 (1.02)

2012), AFG-k-means (Gan & Ng, 2015), and LARFDSSOM. The Adjusted Rand Index (ARI) was used for assessing performances. Table 4 shows that LARFDSSOM2 significantly outperforms all other models in terms of accuracy.

In order to assess the performance of LARFDSSOM2 with high-dimensional gene expression data, three datasets were used, including Leukemia (Statnikov et al., 2005), Prostate Cancer (Yu et al., 2008), and Brain_Tumor2 (Statnikov et al., 2005). They are characterized by high-dimensional samples ranging from 6,033 to 10,367 in up to four categories, and few samples. The results are shown in Table 5. We compared the proposed method with SWCC and DHLRS. As we can see in bold figures, LARFDSSOM2 outperforms the other models.

4.4. Multi-view experiments

The performance of LARFDSSOM2 in multi-view data was assessed with both Normalized Mutual Information (NMI) and Clustering Accuracy (ACC). In this case, we compared LARFDSSOM2 with S Spectral and S Subspace (Shi & Malik, 2000), PairwiseSC and CentroidSC (Kumar et al., 2011), Co-Training (Kumar & Daumé, 2011), Multi NMF (Liu et al., 2013), Multi CF (Wang et al.,

Table 6

Clustering results (ACC) on four datasets by ten algorithms. Numbers in parenthesis are the standard deviations. Bold numbers indicate the best results.

ACC (%)	Digits	Movies617	WebKB	Animal
S Spectral	66.37(4.44)	25.70(1.13)	56.79(1.25)	27.21(1.50)
S Subspace	76.60(4.78)	28.56(1.15)	64.79(6.61)	28.11(1.36)
PairwiseSC	80.82(6.30)	27.89(1.64)	51.63(1.76)	31.65(1.59)
CentroidSC	82.77(7.14)	28.57(1.17)	53.90(2.31)	31.06(2.02)
Co Training	80.22(6.84)	30.74(1.28)	54.25(2.70)	30.35(1.48)
Multi NMF	69.24(6.28)	26.99(1.19)	*	30.56(1.02)
Multi CF	72.45(4.10)	29.60(1.10)	59.79(0.33)	32.11(1.86)
MultiSSC	86.16(5.93)	33.13(1.18)	65.53(1.03)	32.51(1.35)
LARFDSSOM	87.12 (1.07)	36.76 (1.12)	71.85 (1.18)	33.78 (1.02)
LARFDSSOM2	91.05 (1.76)	47.32 (1.68)	77.53 (1.34)	34.03 (2.21)

Table 7

Clustering results (NMI) on four datasets by ten algorithms. Numbers in parenthesis are the standard deviations. Bold numbers indicate the best results.

NMI (%)	Digits	Movies617	WebKB	Animal
S Spectral	62.30(1.85)	25.47(0.85)	40.53(1.91)	15.70(0.65)
S Subspace	71.69(1.96)	29.37(1.01)	42.58(3.45)	17.36(1.06)
PairwiseSC	75.84(2.37)	28.04(0.73)	38.54(0.69)	19.90(1.51)
CentroidSC	76.76(2.58)	28.02(0.72)	38.17(1.05)	18.50(1.48)
Co Training	75.90(2.27)	30.74(1.28)	36.24(1.74)	18.98(0.73)
Multi NMF	65.05(2.30)	27.45(0.55)	*	18.77(0.71)
Multi CF	74.55(2.49)	30.09(1.32)	32.94(3.39)	21.25(1.76)
MultiSSC	83.30(3.75)	33.37(0.98)	45.01(3.51)	21.28(0.57)
LARFDSSOM	84.97 (1.49)	35.27 (1.33)	52.56(2.58)	26.57 (1.56)
LARFDSSOM2	87.44 (2.91)	38.28 (2.35)	53.05 (1.77)	27.98 (1.61)

2013), MultiSSC (Yin et al., 2015), and LARFDSSOM (Bassani & Araújo, 2015).

Table 6 shows the average accuracy and standard deviations of different algorithms for four datasets: Digits (Dua & Graff, 2017), Movies617, WebKB,¹ and Animal. Table 7 shows NMI averages and standard deviations for the same datasets. Regarding the ACC measure, it is worth noting the performance improvement of LARFDSSOM2 for sparse-sample datasets, such as Movies617 and WebKB. In Table 7, LARFDSSOM2 has shown resilience when a clustering quality metric other than accuracy is used.

LARFDSSOM2 was also compared with other more recent models (Section 2.2) that employ subspace clustering to handle multi-view problems. Hence, iLMSC and gLMSC (Zhang et al., 2020), t-SVD-MS (Xie et al., 2018), ETLMS (Wu et al., 2019), and FCMSC and gr-FCMSC (Zheng et al., 2020) were considered. Four new datasets BBC-Sport (Greene & Cunningham, 2006), Caltech101 (Fei-Fei et al., 2004), MITIndoor-67 (Quattoni & Torralba, 2009), and Scene-15 (Fei-Fei & Perona, 2005) were used. Experiments with LARFDSSOM and LARFDSSOM2 used the same feature extractors of their original papers.

The proposed model showed competitive performance in all cases, as shown in Table 8 in which the best results are the bold numbers. LARFDSSOM2 achieves the best overall results under different metrics (ACC, NMI, and ARI) on the datasets BBC-Sport, Caltech101, Scene-15, and MITIndoor-67.

4.5. Computer vision experiments

LARFDSSOM2 was tested with four different sets of computer vision data, including the motion segmentation with the dataset Hopkins 155, the Extended Yale B faces (Yale, 2001), Columbia University Image Library (COIL-20) (Nene et al., 1996), and the data from the Canadian Institute for Advanced Research (CIFAR-10) (Krizhevsky & Hinton, 2009). This section is organized in two parts: motion segmentation and image clustering experiments.

4.5.1. Motion segmentation experiments

Motion segmentation is an important application of subspace clustering that aims to cluster a set of tracking points into different clusters in a way that each cluster corresponds to one motion. The Hopkins 155 contains video sequences that belong to three different categories: checkerboard, articulated, and traffic. In each sequence two or three motions are shown, so we can compare using different methods with two, three, and all motions.

LARFDSSOM2, applied to the Hopkins 155 Dataset² was evaluated using the *Clustering Error*. We compared the following algorithms with LARFDSSOM2: RANSAC (Fischler & Bolles, 1981), MSL (Sugaya & Kanatani, 2004), GPCA (Vidal et al., 2005), LSA (Yan & Pollefeys, 2006), ALC (Ma et al., 2007), SSC (Elhamifar & Vidal, 2009), and LARFDSSOM (Bassani & Araújo, 2015).

Tables 9, 10, and 11 show the error mean and median of LARFDSSOM2 and the other algorithms. Table 9 consists of all results of video sequences with only two moving objects while Table 10 consists of all results with three moving objects in the scene. In both tables, the results consider Checkerboard, Traffic, and Articulated sequences. Table 11 shows the average overall error of 155 Hopkins dataset video sequences. For videos with 2 moving objects, LARFDSSOM2 surpasses all other competitors.

LARFDSSOM2 obtained a lower error rate than all sets of 3-moving object videos (Table 10); the average error rate was 2.00%, against the SSC which had a 2.45% error rate. Finally, Table 11 ratifies the best performance of LARFDSSOM2 and the poor performance of LARFDSSOM.

4.5.2. Image clustering experiments

The Extended Yale B Faces dataset contains 2,414 frontal face images of 38 persons in 64 different lighting conditions. We carried out several types of tests. The first one was with the entire dataset in which each original 192×168 image was resized to a 48×42 image. Then, each image was vectorized to obtain a 2,016-dimensional data vector. In the other experiments, we considered some feature extractors to compare the results with other approaches.

The Columbia University Image Library (COIL-20 dataset) consists of a set of 72 grayscale images for each one of the 20 different rotating objects. Each 128×128 image was acquired at different angles i.e. rotated every 5 degrees. Like the previous set of experiments, the images were vectorized to obtain 256-dimensional data vectors, so there are $n = 1,440$ samples.

CIFAR-10 is a dataset formed by a set of 60,000 color images of 32×32 size. It has 10 categories with 6,000 instances per category. The original RGB images were also vectorized, resulting in 60,000 samples of 3,072 dimensions ($32 \times 32 \times 3$) vectors.

In the experiments, we use raw data and data yielded by the following feature extractors: dense scale-invariant feature transform (DSIFT) (Lowe, 2004), square root of DSIFT (SDSIFT) (Wolf et al., 2009), and the histogram of oriented gradients (HOG) (Dalal & Triggs, 2005). In sum, we carry out experiments with LARFDSSOM and LARFDSSOM2 applied to Extended Yale B Faces, COIL-20, and CIFAR-10 with DSIFT, SDSIFT and HOG feature extractors to compare with a number of state-of-the-art methods.

We divide each image into multiple patches to apply DSIFT and SDSIFT producing the samples for each patch of size 12×12 for Extended Yale B Faces and 4×4 for CIFAR-10 and COIL-20. Then, the outputs of the SIFT-based descriptors were concatenated to produce 32,256 and 2048-dimensional feature vectors for Extended Yale B faces, CIFAR-10, and COIL-20. For HOG, each image was divided into multiple blocks of 4×4 for each dataset. Then, a 9-dimensional HOG feature vector is extracted from each block. After concatenating the features of all patches of each image, the dimension is 16,250 for Extended Yale B Faces and 2,880 for COIL-20 and CIFAR-10.

NMI, ACC, and ARI were used to evaluate the three datasets. The algorithms chosen for comparison are Struct AE (Peng et al., 2018), SSC (Majumdar, 2018), and SDEC (Ren et al., 2019). Such algorithms employ deep learning to categorize data. They were tested for a number of data sets, in particular, image categorization employing the Extended Yale B faces, Coil-20, and CIFAR-10 datasets. Struct AE, SSC, and SDEC are supervised methods, so they do not perform subspace clustering; however, they categorize high-dimensional data. We included the deep-learning models aiming to compare our proposal with a very effective approach, despite those methods having high computational cost.

After performing all the tests, the results suggest that LARFDSSOM2 was more effective than the deep-learning algorithms. Our algorithm outperforms in two (Coil-20 and CIFAR-10) out of the three datasets in terms of the three evaluation metrics. In such experiments, we use feature descriptors instead of raw data for testing all datasets. Thus, SDSIFT was the descriptor that delivered the best results (Tables 13 and 14).

LARFDSSOM2 also was effective with the Extended Yale B faces (Table 12), however, less than with the deep methods. The best average accuracy value is 90.33% with SDSIFT and is 4.37% shorter than the best result obtained by Struct AE. It is also worth mentioning that LARFDSSOM2 performs effectively when the raw data is used; in this case, the average accuracy value is 89.94%. The most common cause of error is when the lighting condition is bad and does not allow the face to be recognized directly. In those cases the pixel values between those kinds of images are really similar, so, when calculating the activation function of

¹ <http://membres-lig.imag.fr/grimal/data.html>

² <http://www.vision.jhu.edu/data/hopkins155/>

Table 8

Clustering results in four datasets by six algorithms. Numbers in parenthesis are the standard deviations. Bold numbers indicate the best results.

Dataset	Method	ACC	NMI	ARI
BBC Sport	ILMSC	91.07 (0.59)	82.59 (0.81)	–
	gLMSC	96.32 (0.78)	88.66 (0.46)	–
	FCMSC	96.51 (0.00)	89.04 (00.00)	–
	gr-FCMSC	96.70 (0.00)	89.73 (0.00)	–
	t-SVD-MSC	94.10	83.00	85.30
	ETLMSC	97.80	98.4	96.7
	LARFDSSOM	92.63(0.2)	95.12(0.61)	90.91(0.21)
Caltech101	LARFDSSOM-2	97.20(0.54)	99.11(0.14)	97.28(0.16)
	ILMSC	52.84 (1.30)	76.26 (1.11)	–
	gLMSC	59.68 (0.60)	81.63 (1.10)	–
	t-SVD-MSC	60.70 (0.5)	85.80 (0.3)	43.00 (0.5)
	ETLMSC	63.9	89.9	45.6
	LARFDSSOM	62.36(1.22)	83.08(1.33)	41.2(0.02)
	LARFDSSOM-2	65.41(0.89)	87.41(0.42)	49.6(0.01)
MIT Indoor-67	t-SVD-MSC	68.40 (0.5)	75.00 (0.7)	55.50 (0.5)
	ETLMSC	77.50	89.9	72.9
	LARFDSSOM	75.82(1.06)	82.36(2.36)	66.45(1.16)
	LARFDSSOM-2	79.63(1.12)	90.02(1.36)	74.26(1.02)
Scene-15	t-SVD-MSC	81.2 (0.7)	85.80 (0.7)	77.10 (0.3)
	ETLMSC	87.80	90.20	85.10
	LARFDSSOM	83.6(0.51)	82.17(0.15)	80.20(0.1)
	LARFDSSOM-2	89.07(0.2)	87.2(0.3)	89.20(0.1)

Table 9

Clustering results (ERR) on four video datasets (2-moving objects) by eight algorithms. Standard deviations in parenthesis. Bold numbers are the best results.

ERR (%)	Checkerboard	Traffic	Articulated	All
GPCA	6.09(1.03)	1.41(0.00)	2.88(0.00)	4.59(0.38)
LSA	2.57(0.27)	5.43(1.48)	4.10(1.22)	3.45(0.59)
RANSAC	6.52(1.75)	2.55(0.21)	7.25(2.64)	5.56(1.18)
MSL	4.46(0.00)	2.23(0.00)	7.23(0.00)	4.14(0.00)
ALC	1.55(0.29)	1.59(1.17)	10.70(0.95)	2.40(0.43)
SSC	0.83(0.00)	0.02(0.00)	0.62(0.00)	0.75(0.00)
LARFDSSOM	5.18 (0.03)	3.72 (0.01)	4.23 (0.02)	2.73 (0.01)
LARFDSSOM2	0.81(0.00)	0.01(0.00)	0.27(0.00)	0.71(0.00)

Table 10

Clustering results (ERR) on four datasets by eight algorithms. Video sequences with three moving objects. Numbers in parenthesis are the standard deviations. Bold numbers indicate the best results.

ERR (%)	Checkerboard	Traffic	Articulated	All
GPCA	31.95(32.93)	19.83(19.55)	16.85(16.85)	28.66(28.26)
LSA	5.80(1.77)	25.07(23.79)	7.25(7.25)	9.73(2.33)
RANSAC	25.78(26.00)	12.83(11.45)	21.38(21.38)	22.94(22.03)
MSL	10.38(4.61)	1.80(0.00)	2.71(2.71)	8.23(1.76)
ALC	5.20(0.67)	7.75(0.49)	21.08(21.08)	6.69(0.67)
SSC	2.97(0.27)	0.58(0.00)	1.42(0.00)	2.45(0.20)
LARFDSSOM	21.23 (1.47)	13.42 (2.12)	14.25 (1.56)	12.32 (1.34)
LARFDSSOM2	2.36(0.09)	0.56(0.00)	1.06(0.00)	2.00(0.15)

Table 11

Clustering results (ERR) on the complete Hopkins data. Bold numbers indicate the best results. LF and LF2 stand for LARFDSSOM and LARFDSSOM2.

ERR (%)	GPCA	LSA	RANSAC	MSL	ALC	SSC	LF	LF2
Mean	10.34	4.94	9.76	5.03	3.56	1.45	8.43	1.01
Median	2.54	0.9	3.21	0.0	0.5	0.0	0.0	0.0

the nodes, the values between the resulting neighboring nodes are really close. Therefore, the resulting winning node is a node

that is close to the true winning one on the map. When dealing with the face clustering problem, LARFDSSOM2 was not able to

Table 12

Clustering results (ACC, NMI, ARI) on Yale B dataset. Numbers in parenthesis are the standard deviations. Bold numbers indicate the best results.

Method	Feature	ACC	NMI	ARI
STRUCT AE	SDSIFT	94.70(2.50)	96.58(0.76)	92.29(2.44)
SSC	DSIFT	91.75(1.58)	93.26(0.28)	85.62(1.00)
	HOG	93.46(2.39)	98.93(0.92)	93.43(2.95)
SSSC	–	55.5(1.26)	60.3(0.29)	–
SSSC(SLRR)	–	68.9(1.19)	74.0(0.60)	–
SSSC(SLSR)	–	59.9(1.45)	65.2(0.61)	–
SC-CNMF1	–	67.08(0.5)	76.22(0.3)	–
SC-CNMF2	–	76.22(0.3)	76.24(0.7)	–
LARFDSSOM	–	63.11(1.45)	65.32(1.23)	70.04(1.12)
	DSIFT	67.24(1.43)	66.78(2.13)	72.34(0.78)
	SDSIFT	70.33(1.22)	69.36(0.78)	74.23(1.15)
	HOG	64.56(1.34)	65.98(1.02)	71.89(1.11)
LARFDSSOM2	–	89.94(0.47)	87.41 (0.13)	80.52(1.36)
	DSIFT	89.79(0.32)	89.08(1.19)	81.23(0.78)
	SDSIFT	90.33(1.02)	91.17(1.34)	82.25(1.08)
	HOG	90.34(0.35)	87.23(0.68)	80.79(1.46)

Table 13

Clustering results (NMI, ARI, ACC) on Coil-20 dataset. Numbers in parenthesis are the standard deviations. Bold numbers indicate the best results.

Method	Feature	ACC	NMI	ARI
STRUCT_AE	SDSIFT	91.04(2.23)	94.67(0.99)	88.61(1.95)
	LPQ	84.38(1.67)	89.72(0.57)	81.51(2.02)
SSC	DSIFT	89.21(3.67)	94.38(0.89)	89.92(3.52)
	HOG	87.91(2.48)	93.37(0.68)	84.24(1.05)
t-SVD-MSC	LBP	83.0 (0.000)	88.4 (0.005)	78.6(0.003)
ETLMSC	LBP	87.7	94.7	86.2
LARFDSSOM	–	82.09(0.65)	83.09(1.12)	77.56(1.34)
	DSIFT	84.22(1.18)	86.89(0.76)	79.22(1.04)
	HOG	83.78(1.09)	84.92(0.56)	78.01(1.06)
	SDSIFT	86.32(1.25)	88.45(1.02)	80.04(1.04)
LARFDSSOM2	–	89.09(1.23)	92.02(1.12)	84.02(0.67)
	DSIFT	91.47(1.01)	94.01(1.03)	86.92(1.09)
	HOG	89.78(0.08)	92.76(1.32)	85.66(1.15)
	SDSIFT	93.17(1.34)	94.89(1.09)	87.45(1.03)

Table 14

Clustering results (NMI, ARI, ACC) on CIFAR-10 dataset. Numbers in parenthesis are the standard deviations. Bold numbers indicate the best results.

Method	Feature	ACC	NMI	ARI
STRUCT AE	DSIFT	18.05 (0.07)	4.63 (0.19)	1.89(0.05)
	HOG	25.92(0.95)	12.85(0.53)	6.62(0.37)
	SDSIFT	18.00 (0.00)	3.99(0.22)	1.56 (0.00)
SDEC	HOG	27.26	17.20	9.48
LARFDSSOM	–	17.44(1.03)	4.65(0.67)	1.45(2.31)
	DSIFT	20.57(0.78)	7.90(0.95)	1.88(0.08)
	HOG	19.09(1.43)	6.89(0.14)	2.89(0.56)
	SDSIFT	17.32(0.32)	4.91(0.11)	2.78(0.43)
LARFDSSOM2	–	30.45(2.12)	10.45(0.56)	3.67(0.64)
	DSIFT	23.87(0.67)	11.98(0.07)	2.97(0.15)
	HOG	32.09(1.65)	11.03(0.13)	4.78(0.05)
	SDSIFT	24.56(1.24)	13.87(0.65)	6.19(0.76)

overcome the state-of-the-art methods due to the nature of the dataset. The Extended Yale B Faces dataset contains face images with different lighting conditions which, after the clustering process, result in certain data assigned to the wrong cluster. The similarities among faces of different subjects and the variation of illumination conditions increase clustering difficulty.

When using SDSIFT in the Coil-20 dataset, for example, its performance is at least 2.30% higher than the second-best result in terms of accuracy. Considering the different feature extractors, the best performance is achieved with HOG for CIFAR-10 reaching 32.09% of accuracy and 13.87% as a normalized mutual information value with SDSIFT.

The test Z was used to know if there was a significant difference between the results of LARFDSSOM2 and the compared algorithms. The significance level was 3%, and the hypothesis that median values are statistically similar was rejected. Considering such a hypothesis test, we can argue that LARFDSSOM2 shows the best overall results.

Although LARFDSSOM2 shows competitive results, we believe there is still room for improvement. For example, in the face clustering problem, we have to deal with different lighting conditions. Background can also lead to miscategorization, which points to the importance of defining a region of interest when dealing with these kinds of problems.

4.6. Comparison with a non-supervised neural network

In this section we compare LARFDSSOM2 with a non-supervised Neural Network called Saliency-aware Adaptive Resonance Theory for Large-scale Sparse Data Clustering (SA-ART) (Meng et al., 2019) which is a robust soft subspace clustering approach for sparse data. SA-ART uses three statistical measures, i.e. frequency, mean, and variance, to estimate the importance and distributions of features for each cluster. In such a way, this method incrementally reveals the frequent features that have high and stable values in the cluster, called salient features. Two datasets were used for the experiments: ImageNet (Russakovsky et al., 2015) and BlogCatalog (Wang et al., 2010). They are characterized for having simultaneously high number of samples, features, and categories. The feature extractions we used are the same as those in the original paper. The parameters used for these datasets are the same used in the experiments before, i.e. the same as computer vision for Imagenet and the same used in BBCSport for Blog Catalog regarding to the context of the application.

Table 15 shows the results of LARFDSSOM2 over the datasets mentioned above. The metrics used in this case are: precision (P), weighted average recall (R), and F1 score (F). Five runs were conducted in order to follow the same procedure as SA-ART. LARFDSSOM2 achieve the best results under all these different metrics.

4.7. Parametric adjustments

LARFDSSOM2 has seven out of ten parameters adjusted by using Latin Hypercube Sampling (LHS) (Chatterjee & Hadi, 2009). Then, we ran LARFDSSOM2 500 times for distinct parameter setups extracted from their valid interval. We also determined the most relevant parameters among them all; changes in a_t and l_p , as in Bassani and Araújo (2015) may cause the most significant changes in behavior. Three parameters were not tuned through LHS; t_{max} , $maxcom$, and N_{max} were set to make LARFDSSOM2 stop running when the clustering converges.

We tuned seven sets of parameters regarding the type of datasets: UCI single view, gene expression, multi-view, motion segmentation, extended Yale B faces, Coil-20, and CIFAR-10. Unlike the other groups of data, the experiments with computer vision demanded a parameter setting for each data set, as do the papers used for comparison. Table 16 shows the setups for raw datasets for all the experiments. Table 17 shows the parameters for multi-view data, while Table 18 presents the parameter values for computer vision sets that have feature extraction processing.

The parameter setup procedure, using LHS, suggests that LARFDSSOM2 presented (i) high sensibility to modifications of a_t in all cases and it should remain high (close to 1); (ii) high sensibility to modifications of l_p in all cases and it should remain low (close to 0); (iii) low sensibility to modifications of the remaining parameters.

5. Discussion and conclusion

LARFDSSOM and LARFDSSOM2 are both examples of soft subspace clustering (SSC). The former often establishes few relevancies for which values are different from zero or one, standing farther from a soft subspace clustering algorithm than LARFDSSOM2. Our proposal has clear soft subspace clustering behavior, determining continuous values for relevancies ranging from zero to one for cluster separation. This is an influential reason why LARFDSSOM2 does better than LARFDSSOM for all cases (Tables 3–15). Moreover, LARFDSSOM2 is an independent SSC instance, yielding a model that does not need to pre-establish the number of clusters; it considers different levels of similarity of samples for clustering, and determines neighborhoods between clusters based on the proximity of the relevance vectors, the dimensions that are relevant to form each cluster.

LARFDSSOM2 performs feature weighting to assess the relative prominence of each attribute with respect to each category and to assign a weight to it. A relevant attribute to characterize a cluster is weighted with higher relevance values than those of non-influential features. LARFDSSOM2 weighting uses continuous values and thus there is a finer tuning to determine relevance. The use of weighting attributes is seen as a more precise technique to choose important attributes in tasks where the features present distinct relevancies (Wettschereck et al., 1997) as usual in many real-world problems. Relevancies for attribute/dimension can guide the clustering process so that the data are partitioned into more meaningful subsets and they can be used in subsequent steps of the learning system to improve data categorization.

Experimentally, we verified that to learn relevance, ω_{ji} , we must take into consideration the component change δ_{ji} over time. In LARFDSSOM, the relevance vector does not change much over time and its slope is steep even when using a smoothing factor. LARFDSSOM2 has a mechanism for more frequent and subtle alterations of the relevance vector. It updates the relevance vector of the most active nodes, the winner node and its neighbors, in a similar way to weight updating for SOM. The relevancies in LARFDSSOM2 are calculated based on the median absolute deviation (MAD), which differs from the first version of the algorithm. As a consequence, the distribution of the relevance values has changed significantly. After training LARFDSSOM, the relevance values had the following distribution: 25% of values equal to zero, 47% of values greater than zero and less than one, and 28% of values equal to one. On the other hand, LARFDSSOM2 presented a lot more values within the central interval: 19% of values equal to zero, 73% of values greater than zero and less than one, and 8% of values equal to one. This result suggests that LARFDSSOM2 considers the intermediate level of relevance of a higher number of dimensions.

LARFDSSOM2 is capable of discovering latent sub-clusters, a welcome feature in a data mining context (knowledge extraction). Such subcategories are fine subdivisions of the categorization, depending on the parameter tuning, in particular of the value for the activation threshold. The remaining nodes in the resultant map represent the clusters, although, more than one node can represent subcategories of a category, usually, neighbors in the map. Therefore, the relevance vectors of the subcategories are similar, a straight implication of being neighbors. For example, the Coil-20 dataset has 20 categories; therefore, the final map should have at least 20 remaining nodes. After a number of experiments, the number of nodes forming the map was greater than the number of desired categories. For this case, there were 32 remaining nodes in the final map, so in some cases, two nodes represent the same class, i.e. the second node is considered as a subcategory of the first one.

Likewise, in the dataset Vehicle, which has 18 dimensions and 4 clusters, the number of remaining nodes in the final map

Table 15

Clustering results (P, R e F) on Imagenet and BlogCatalog datasets. Numbers in parenthesis are the standard deviations. Bold numbers indicate the best results.

Dataset	Method	P	R	F
BlogCatalog	SA-ART	64.5(2.6)	17.6(2.8)	33.7(2.2)
	LARFDSSOM	66.21(2.3)	19.54(3.6)	35.12(1.2)
	LARFDSSOM2	70.12(1.05)	23.11(1.1)	39.05(1.13)
Imagenet	SA-ART	44.9(3.3)	12.9(3.6)	25.2(3.7)
	LARFDSSOM	47.23(2.1)	16.9(1.4)	29.36(1.1)
	LARFDSSOM2	52.70(1.2)	19.71(2.2)	32.27(1.47)

Table 16

Parameters of LARFDSSOM2 for each dataset considering raw data.

Parameter	UCI	Gene expression	Multi-view	Motion segmentation	Extended Yale B faces	Coil-20	CIFAR-10
e_b	0.05	0.5	0.05	0.01	0.05	0.05	0.05
e_n	0.001	0.001	0.01	0.001	0.01	0.01	0.001
β	0.5	0.8	0.1	0.5	0.5	0.5	0.5
s_l	0.01	0.1	0.001	0.1	0.01	0.01	0.01
c	0.5	0.05	0.5	0.5	0.05	0.5	0.5
t_{max}	100	30	50	50	50	50	50
$maxcomp$	100	30	50	50	30	30	50
N_{max}	15	40	10	20	20	20	20
a_t	0.978	0.99	0.978	0.99	0.995	0.97712	0.97812
l_p	0.001	0.0005	0.001	0.001	0.01	0.01	0.01

Table 17

Parameters of LARFDSSOM2 for each Multi-view dataset considering raw data.

Parámetros	Caltech101	MIT Indoor-67	Scene-15	BBC Sport
e_b	0.05	0.05	0.05	0.05
e_n	0.01	0.001	0.001	0.01
β	0.5	0.5	0.5	0.1
s	0.01	0.01	0.01	0.01
c	0.5	0.5	0.5	0.5
t_{max}	50	50	40	50
$maxcomp$	50	30	50	50
N_{max}	20	40	20	10
a_t	0.995	0.9956	0.9978	0.978
l_p	0.001	0.001	0.001	0.0005

Table 18

Parameters of LARFDSSOM2 for computer vision datasets using feature descriptors - DSIFT, SDSIFT and HOG.

Parameter	E Yale B			COIL-20			CIFAR-10		
	DSIFT	SDSIFT	HOG	DSIFT	SDSIFT	HOG	DSIFT	SDSIFT	HOG
e_b	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
e_n	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
β	0.5	0.5	0.8	0.5	0.5	0.5	0.5	0.5	0.5
s_l	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
c	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
t_{max}	40	40	40	50	50	50	50	50	50
$maxcomp$	50	50	50	30	30	30	30	30	30
N_{max}	20	20	20	20	20	20	20	20	20
a_t	0.998	0.9956	0.9978	0.97856	0.97812	0.9978	0.97856	0.97856	0.97856
l_p	0.01	0.01	0.01	0.01	0.01	0.001	0.01	0.01	0.01

differs a little from the desired categories, so that means the existence of related subcategories forming a cluster with the closest nodes. Fig. 2 shows a projection in two dimensions of such a dataset, there are 7 remaining nodes in the map determined by LARFDSSOM2.

Another important conclusion is that a very high N_{max} might produce disconnected small clusters (nodes without links to other

nodes), a potential reason for decreasing the overall clustering accuracy. This parameter setting must avoid creating lookup tables and must not group similar elements.

One of the main advantages of LARFDSSOM2 is its good performance dealing with high-dimensional datasets. This kind of data is normally more difficult to work with due to its large number of features or dimensions. LARFDSSOM2 presented significant

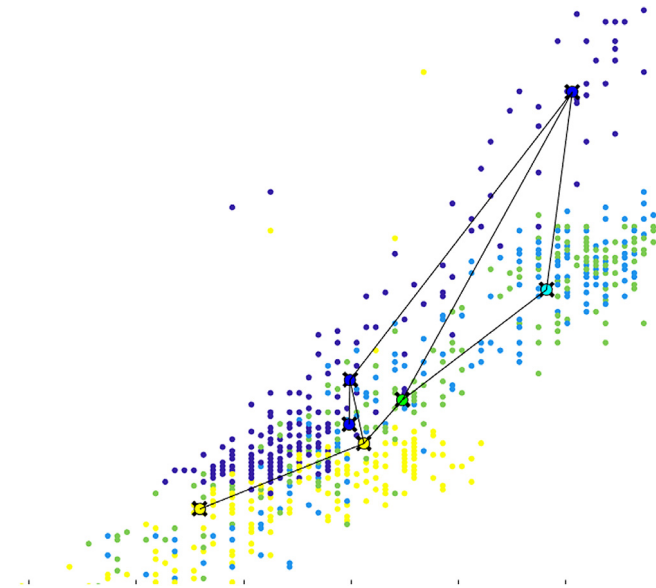


Fig. 2. Projection in two dimensions of a real dataset (Vehicle) and the nodes determined by LARFDSSOM2.

performance improvement when working with gene expression data, overcoming the results of the compared algorithms. We also tested the algorithm with computer vision problems; our method performs well when dealing with face clustering. We compare favorably with algorithms that use deep neural networks (DNN), a family of high-performance supervised models, and we get close performances in terms of quality measurements; sometimes LARFDSSOM2 does better and sometimes it does worse than the DNN-based models. When we compare LARFDSSOM2 with low-rank representation and sparse methods, our results suggest significant efficacy.

One of the focuses of this work is data from multiple views, including BBCSport, Scene-15, Caltech-101, and MIT Indoor-67. With the experimental results, we can see that LARFDSSOM2 overcome almost all the comparison methods, which objective is to deal just with multi-view data, meanwhile, LARFDSSOM2 deals with various problems and performed really well in all of them.

We tested clustering for the sparse high-dimensional datasets Movies617 and WebKB, common types of data in real world problems in which the vast majority of attributes are zero valued. Very often, this is a consequence of transforming categorical variables to binary values or converting text to an orderless representation such as bag-of-words. On the other hand, the performance for the Animal dataset has plenty of room to improve, especially because of its size. This was a constraint, especially when tuning the parameters in the search for accuracy. Also, there is room for improvement for the computer vision datasets, especially for CIFAR-10. Studies to tackle these limitations can be very relevant. Moreover, to accelerate the LARFDSSOM2 processing, a massively parallel implementation of it, with GPUs or multicore CPUs using CUDA, is a potential alternative for future research.

Declaration of competing interest

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

Acknowledgments

This work was partially supported by CAPES, Brazil (Cooperação de Aperfeiçoamento de Pessoal de Nível Superior) a research agency in Brazil.

References

- Aggarwal, C. C., Hinneburg, A., & Keim, D. A. (2001). On the surprising behavior of distance metrics in high dimensional space. In *Proceeding of the international conference on database theory* (pp. 420–434). Springer.
- Alizadeh, A. A., Eisen, M. B., Davis, R. E., Ma, C., Lossos, I. S., Rosenwald, A., Boldrick, J. C., Sabet, H., Tran, T., Yu, X., Powell, J. I., Yang, L., Marti, G. E., Moore, T., Hudson, J., Lu, L., Lewis, D. B., Tibshirani, R., Sherlock, G., ... Staudt, L. M. (2000). Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature*, 403(6769), 503–511. <http://dx.doi.org/10.1038/35000501>.
- Araújo, A. F. R., & Rego, R. L. (2013). Self-organizing maps with a time-varying structure. *ACM Computing Surveys*, 46(1), 7:1–7:38.
- Armstrong, S. A., Staunton, J. E., Silverman, L. B., Pieters, R., den Boer, M. L., Minden, M. D., Sallan, S. E., Lander, E. S., Golub, T. R., & Korsmeyer, S. J. (2002). MLL translocations specify a distinct gene expression profile that distinguishes a unique leukemia. *Nature Genetics*, 30(1), 41–47.
- Bassani, H. F., & Araújo, A. F. R. (2012). Dimension selective self-organizing maps for clustering high dimensional data. In *Neural networks (IJCNN), the 2012 international joint conference on IEEE* (pp. 1–8).
- Bassani, H. F., & Araújo, A. F. R. (2015). Dimension selective self-organizing maps with time-varying structure for subspace and projected clustering. *IEEE Transactions on Neural Networks and Learning Systems*, 26(3), 458–471.
- Bassani, H. F., & Araújo, A. F. R. (2019). A neural network architecture for learning word-referent associations in multiple contexts. *Neural Networks*, 117, 249–267.
- Chan, E. Y., Ching, W. K., Ng, M. K., & Huang, J. Z. (2004). An optimization algorithm for clustering using weighted dissimilarity measures. *Pattern Recognition*, 37(5), 943–952.
- Chatterjee, S., & Hadi, A. S. (2009). Sensitivity analysis in linear regression, 327. John Wiley & Sons.
- Chen, G., & Lerman, G. (2009). Spectral curvature clustering (SCC). *International Journal of Computer Vision*, 81(3), 317–330.
- Chen, X., Ye, Y., Xu, X., & Huang, J. Z. (2012). A feature group weighting method for subspace clustering of high-dimensional data. *Pattern Recognition*, 45(1), 434–446.
- Chowdary, D., Lathrop, J., Skelton, J., Curtin, K., Briggs, T., Zhang, Y., Yu, J., Wang, Y., & Mazumder, A. (2006). Prognostic gene expression signatures can be measured in tissues collected in RNAlater preservative. *The Journal of Molecular Diagnostics*, 8(1), 31–39.
- Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection.
- Deng, Z., Choi, K.-S., Jiang, Y., Wang, J., & Wang, S. (2016). A survey on soft subspace clustering. *Information Sciences*, 348, 84–106.
- Domeniconi, C., Papadopoulos, D., Gunopulos, D., & Ma, S. (2004). Subspace clustering of high dimensional data. In *Proceedings of the 2004 SIAM international conference on data mining* (pp. 517–521).
- Dua, D., & Graff, C. (2017). *UCI machine learning repository*. University of California, Irvine, School of Information and Computer Sciences, URL: <http://archive.ics.uci.edu/ml>.
- Elhamifar, E., & Vidal, R. (2009). Sparse subspace clustering. In *2009 IEEE conference on computer vision and pattern recognition* (pp. 2790–2797).
- Elhamifar, E., & Vidal, R. (2013). Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11), 2765–2781.
- Fei-Fei, R., Fergus, R., & Perona, P. (2004). Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop* (pp. 178–178).
- Fei-Fei, L., & Perona, P. (2005). 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).
- Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6), 381–395.
- Gan, G., & Ng, M. K.-P. (2015). Subspace clustering with automatic feature grouping. *Pattern Recognition*, 48(11), 3703–3713.
- Gan, G., & Wu, J. (2008). A convergence theorem for the fuzzy subspace clustering (FSC) algorithm. *Pattern Recognition*, 41(6), 1939–1947.
- Gordon, G. J., Jensen, R. V., Hsiao, L.-L., Gullans, S. R., Blumenstock, J. E., Ramaswamy, S., Richards, W. G., Sugarbaker, D. J., & Bueno, R. (2002). Translation of microarray data into clinically relevant cancer diagnostic tests using gene expression ratios in lung cancer and mesothelioma. *Cancer Research*, 62(17), 4963–4967.

- Greene, D., & Cunningham, P. (2006). Practical solutions to the problem of diagonal dominance in kernel document clustering. In *Proc. 23rd international conference on machine learning* (pp. 377–384). ACM Press.
- Hammer, B., & Villmann, T. (2002). Generalized relevance learning vector quantization. *Neural Networks*, 15(8), 1059–1068.
- He, R., Wang, L., Sun, Z., Zhang, Y., & Li, B. (2015). Information theoretic subspace clustering. *IEEE Transactions on Neural Networks and Learning Systems*, 27(12), 2643–2655.
- Jing, L., Ng, M. K., & Huang, J. Z. (2007). An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data. *IEEE Transactions on Knowledge and Data Engineering*, 19(8), 1026–1041.
- Jing, L., Ng, M. K., Xu, J., & Huang, J. Z. (2005). Subspace clustering of text documents with feature weighting k-means algorithm. In *Proceedings of the 9th Pacific-Asia conference on knowledge discovery and data mining* (pp. 802–812). Springer.
- Kelkar, B. A., & Rodd, S. F. (2019). Subspace clustering—A survey. In *Data management, analytics and innovation* (pp. 209–220). Springer.
- Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE*, 78(9), 1464–1480.
- Kriegel, H.-P., Kröger, P., & Zimek, A. (2009). Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 3(1), 1:1–1:58.
- Krizhevsky, A., & Hinton, G. (2009). *Learning multiple layers of features from tiny images: Technical report*. Citeseer.
- Kumar, A., & Daumé, H. (2011). A co-training approach for multi-view spectral clustering. In *Proceedings of the 28th international conference on machine learning* (pp. 393–400).
- Kumar, A., Rai, P., & Daume, H. (2011). Co-regularized multi-view spectral clustering. In *Advances in neural information processing systems* (pp. 1413–1421).
- Li, X., Lu, Q., Dong, Y., & Tao, D. (2018). Robust subspace clustering by cauchy loss function. *IEEE Transactions on Neural Networks and Learning Systems*.
- Liu, J., Cheng, Y., Wang, X., Cui, X., Kong, Y., & Du, J. (2018a). Low rank subspace clustering via discrete constraint and hypergraph regularization for tumor molecular pattern discovery. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 15(5), 1500–1512.
- Liu, J., Cheng, Y., Wang, X., Cui, X., Kong, Y., & Du, J. (2018b). Low rank subspace clustering via discrete constraint and hypergraph regularization for tumor molecular pattern discovery. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 15(5), 1500–1512.
- Liu, S., Maljovec, D., Wang, B., Bremer, P.-T., & Pascucci, V. (2016). Visualizing high-dimensional data: Advances in the past decade. *IEEE Transactions on Visualization and Computer Graphics*, 23(3), 1249–1268.
- Liu, J., Wang, C., Gao, J., & Han, J. (2013). Multi-view clustering via joint non-negative matrix factorization. In *Proceedings of the 2013 SIAM international conference on data mining* (pp. 252–260). SIAM.
- Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2), 91–110.
- Ma, Y., Derksen, H., Hong, W., & Wright, J. (2007). Segmentation of multivariate mixed data via lossy data coding and compression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(9), 1546–1562.
- Majumdar, A. (2018). Graph structured autoencoder. *Neural Networks*, 106, 271–280.
- Medeiros, H. R., de Oliveira, F. D., Bassani, H. F., & Araújo, A. F. R. (2019). Dynamic topology and relevance learning SOM-based algorithm for image clustering tasks. *Computer Vision and Image Understanding*, 179, 19–30.
- Meng, L., Tan, A.-H., & Miao, C. (2019). Saliency-aware adaptive resonance theory for large-scale sparse data clustering. *Neural Networks*, 120, 143–157, Special issue in honor of the 80th birthday of Stephen Grossberg.
- Nene, S., Nayar, S., & Murase, H. (1996). Columbia university image library (coil-20), (1996-02) [2010-05-23]. <http://www1.cs.columbia.edu/CA-VE/software/softlib/coil-20.php>.
- Parsons, L., Haque, E., & Liu, H. (2004). Subspace clustering for high dimensional data: a review. *ACM SIGKDD Explorations Newsletter*, 6(1), 90–105.
- Peng, X., Feng, J., Xiao, S., Yau, W.-Y., Zhou, J. T., & Yang, S. (2018). Structured autoencoders for subspace clustering. *IEEE Transactions on Image Processing*, 27(10), 5076–5086.
- Peng, X., Tang, H., Zhang, L., Yi, Z., & Xiao, S. (2015). A unified framework for representation-based subspace clustering of out-of-sample and large-scale data. *IEEE Transactions on Neural Networks and Learning Systems*, 27(12), 2499–2512.
- Pereira, R. B., Plastino, A., Zadrozny, B., & Merschmann, L. H. (2018). Categorizing feature selection methods for multi-label classification. *Artificial Intelligence Review*, 49(1), 57–78.
- Quattoni, A., & Torralba, A. (2009). Recognizing indoor scenes. In *2009 IEEE conference on computer vision and pattern recognition* (pp. 413–420).
- Ren, Y., Hu, K., Dai, X., Pan, L., Hoi, S. C., & Xu, Z. (2019). Semi-supervised deep embedded clustering. *Neurocomputing*, 325, 121–130.
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., & Bernstein, M. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3), 211–252.
- Saha, M. (2013). A graph based approach to multiview clustering. In *International conference on pattern recognition and machine intelligence* (pp. 128–133). Springer.
- Shi, J., & Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8), 888–905.
- Simiński, K. (2012). Clustering in fuzzy subspaces. *Theoretical and Applied Informatics*, 24(4), 313–326.
- Statnikov, A., Tsamardinos, I., Dosbayev, Y., & Aliferis, C. F. (2005). GEMS: a system for automated cancer diagnosis and biomarker discovery from microarray gene expression data. *International Journal of Medical Informatics*, 74(7–8), 491–503.
- Sugaya, Y., & Kanatani, K. (2004). *Statistical methods in video processing: ECCV 2004 workshop. Revised selected papers* (pp. 13–15). Springer.
- Vidal, R. (2011). Subspace clustering. *IEEE Signal Processing Magazine*, 28(2), 52–68.
- Vidal, R., & Favaro, P. (2014). Low rank subspace clustering (LRSC). *Pattern Recognition Letters*, 43, 47–61.
- Vidal, R., Ma, Y., & Sastry, S. (2005). Generalized principal component analysis (GPCA). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12), 1945–1959.
- Wang, J., Deng, Z., Choi, K.-S., Jiang, Y., Luo, X., Chung, F.-L., & Wang, S. (2016). Distance metric learning for soft subspace clustering in composite kernel space. *Pattern Recognition*, 52, 113–134.
- Wang, H., Nie, F., & Huang, H. (2013). Multi-view clustering and feature learning via structured sparsity. *Proceedings of the 30th international conference on machine learning*, vol. 28 (pp. 352–360).
- Wang, X., Tang, L., Gao, H., & Liu, H. (2010). Discovering overlapping groups in social media. In *2010 IEEE international conference on data mining* (pp. 569–578).
- Wettschereck, D., Aha, D. W., & Mohri, T. (1997). A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms. *Artificial Intelligence Review*, 11(1), 273–314.
- Wolf, L., Hassner, T., & Taigman, Y. (2009). Similarity scores based on background samples. In *Asian conference on computer vision* (pp. 88–97). Springer.
- Wu, J., Lin, Z., & Zha, H. (2019). Essential tensor learning for multi-view spectral clustering. *IEEE Transactions on Image Processing*, 28(12), 5910–5922.
- Xie, Y., Tao, D., Zhang, W., Liu, Y., Zhang, L., & Qu, Y. (2018). On unifying multi-view self-representations for clustering by tensor multi-rank minimization. *International Journal of Computer Vision*, 126(11), 1157–1179.
- Xu, R., & Wunsch I.I., D. C. (2011). BARTMAP: A viable structure for biclustering. *Neural Networks*, 24(7), 709–716.
- Yale (2001). The extended Yale Face Database B. URL <http://vision.ucsd.edu/~iskwak/ExtYaleDatabase/ExtYaleB.html>.
- Yan, J., & Pollefeys, M. (2006). A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In *Proceeding of the European conference on computer vision* (pp. 94–106). Springer.
- Yin, Q., Wu, S., He, R., & Wang, L. (2015). Multi-view clustering via pairwise sparse subspace representation. *Neurocomputing*, 156, 12–21.
- Yu, L., Ding, C., & Loscalzo, S. (2008). Stable feature selection via dense feature groups. In *Proceedings of the 14th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 803–811).
- Zhang, C., Fu, H., Hu, Q., Cao, X., Xie, Y., Tao, D., & Xu, D. (2020). Generalized latent multi-view subspace clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(1), 86–99.
- Zheng, Q., Zhu, J., Li, Z., Pang, S., Wang, J., & Li, Y. (2020). Feature concatenation multi-view subspace clustering. *Neurocomputing*, 379, 89–102.