Journal of Computational Mathematics and Data Science 6 (2023) 100070

Contents lists available at ScienceDirect

Journal of Computational Mathematics and Data Science

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

Classification of hyperspectral images with copulas[☆]

C. Tamborrino [a,*], F. Mazzia [b]

[a] Dipartimento di Matematica, Università degli Studi di Bari Aldo Moro, Italy
[b] Dipartimento di Informatica, Università degli Studi di Bari Aldo Moro, Italy

ARTICLE INFO

Keywords:
Remote sensing
Features extraction
Classification
Copulas

ABSTRACT

In the last decade, supervised learning methods for the classification of remotely sensed images (RSI) have grown significantly, especially for hyper-spectral (HS) images. Recently, deep learning-based approaches have produced encouraging results for the land cover classification of HS images. In particular, the Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) have shown good performance. However, these methods suffer for the problem of the hyperparameter optimization or tuning that requires a high computational cost; moreover, they are sensitive to the number of observations in the learning phase. In this work we propose a novel supervised learning algorithm based on the use of copula functions for the classification of hyperspectral images called CopSCHI (Copula Supervised Classification of Hyperspectral Images). In particular, we start with a dimensionality reduction technique based on Singular Value Decomposition (SVD) in order to extract a small number of relevant features that best preserve the characteristics of the original image. Afterward, we learn the classifier through a dynamic choice of copulas that allows us to identify the distribution of the different classes within the dataset. The use of copulas proves to be a good choice due to their ability to recognize the probability distribution of classes and hence an accurate final classification with low computational cost can be conducted. The proposed approach was tested on two benchmark datasets widely used in literature. The experimental results confirm that CopSCHI outperforms the state-of-the-art methods considered in this paper as competitors.

1. Introduction

In recent years, remote sensing (RS) data has caught a lot of attention. Several space agencies promote satellite mission projects and currently, several satellites gravitate around our planet. On board these satellites there are a number of optical sensors capable of acquiring images with different spatial and spectral resolutions. This type of data, after being properly pre-processed, can be used for different tasks, from change detection to saliency detection to classification of land cover. Given the amount of data available, hyperspectral image analysis (HS) is a highly researched field in the scientific community. These studies have implications in several areas, such as, oceanography, smart city, agriculture [1], health, hazard analysis, military services and vegetation monitoring [2,3]. In machine learning theory, the task of classification is approached with supervised learning techniques and in the context of RS this task is commonly called Land Cover classification. In this direction in the literature, many algorithms have been developed for this purpose; k-nearest-neighbor [4], random forest (RF) [5] and support vector machines (SVM) [6], represent the most traditional methods. Other approaches include, sparse representation (SR), that is an efficient machine-dependent method [7], Markov random

[☆] The research of Francesca Mazzia is funded by PON "Ricerca e Innovazione 2014–2020", project CLOSE "Close to the Earth" (ARS01_00141). The research of Cristiano Tamborrino is funded by PON Project Change Detection in Remote Sensing CUP H94F18000260006.

[*] Corresponding author.
E-mail addresses: cristiano.tamborrino@uniba.it (C. Tamborrino), francesca.mazzia@uniba.it (F. Mazzia).

https://doi.org/10.1016/j.jcmds.2022.100070
Received 10 March 2022; Received in revised form 8 November 2022; Accepted 4 December 2022

field (MRF) [8], extreme learning machine (ELM) [9] and recently, in the field of deep learning, there are several techniques for supervised classification involving neural networks (NN) [10].

An accurate land cover classification by supervised machine learning algorithms requires, during the learning phase, datasets in which the pixels have been appropriately labeled with ground truth information, collected through human intervention *in situ* or with automated techniques. Despite the availability of RS images, this type of information is still limited, especially for HS images. This process requires a lot of effort in terms of time and work, whereas many areas of planet earth are inaccessible and therefore makes it impossible to collect labeled data [11]. Additional methods for supervised classification have been studied based on the combination of spatial and spectral feature extraction techniques [12]. Given an HS image, any spectral vector is referred to as spectral feature, while, spatial feature denotes the spatial relationship between any pixel and its neighbor. In particular, selecting the relevant spectral/spatial features is fundamental to significantly improve the algorithm performance [13]. Several algorithm for features extraction have been developed for the classification of HS images [10,14–16]. An additional consideration in the HS classification is the curse of dimensionality. The accuracy of classifiers is affected by the size of the initial data during the learning phase, which also increases the computational cost. It thus becomes useful to reduce the dimensionality of the dataset, in order to obtain a smaller number of features that keep the more representative information of the considered image. In this context, many methodologies present in the literature, recommend the use of data reduction techniques before the learning phase, such as, Principal Component Analysis (PCA) [17], randomized principal component analysis (R-PCA) [18], Singular Value Decomposition (SVD) [19], Independent Component Analysis (ICA) [20], and Linear Discriminant Analysis, (LDA). In [21] the authors propose a reduction of dimensionality based on the non negative matrix factorization (SSNMF) and the extracted features are used for the unsupervised classification of salient objects in HS images. The same strategy is employed in [22] by using the Autoencoder NN. This type of technique is widely used even when the task is the unsupervised classification, indeed, the addition of spatial information becomes crucial for a better grouping of pixels. In [23] a deep belief network (DBN) that combines (PCA), hierarchical learning-based feature extraction, and logistic regression is employed for unsupervised feature learning classification. In the last decade, in parallel with the well-known machine learning techniques applied in the field of classification, there has been a significant increase in the development of deep learning techniques, through several neural network architectures (NN) [24,25], especially Convolutional Neural Networks (CNN) and Recurrent Neural Network (RNN), that are two of the most common models used to analyze spatial and spectral information. Several studies exist in literature, for instance, an HS image classification framework based on stacked denoise autoencoder (SDAE) is proposed in [26]. A simple but effective CNN architecture containing five layers with weights for supervised HS image classification has been applied in [27], another classification method that hierarchically constructs high level features in an automated way is presented in [18]. Moreover in [28], a technique where, the whole dataset and an advanced architecture with CNN, is presented. Another widely used model of ANN is the RNN that is adopted especially in the classification of data in which temporal information is present, i.e. multi-temporal HS images. This technique can be used in the presence of multi-temporal images. An application for time series satellite images classification using both, RNN and LSTM is presented in [29] and in [30]. In [31], CNN and RNN have also been used in combination: a RNN is recruited to model the dependencies between the features then a CNN is used to learn more discriminative features for HS data classification. The techniques based on deep learning have reached high levels of accuracy in supervised classification. However, some problems still remain open. Among these, the numbers of hyperparameters to tune for the neural network, and the high computational cost needed during the learning stage. [11]. For this reason, in this paper we propose a fully statistical approach for the HS image classification, that combines a Bayesian decision theory and copula functions. Copula functions are suitable tools in statistics for modeling multiple dependence, more in detail, they are able to link the marginal distributions of different variables in order to give a flexible and accurate description of the joint law of the variables of interest [32–34]. Copulas have been studied in many works by the scientific community ranging from finance and economics to hydrology and environmental science [35–38]. However, in the field of RS, and in particular for the classification task, few works that employ copulas have been proposed [39–41]. In this paper, we provide a new contribution for the HS images classification based on a designed algorithm that allows a dynamical choice of the suitable copula after a pre-processing stage in which feature extraction is performed by a matrix factorization. We use different copula functions belonging to two main families, Elliptical and Archimedean. The important step, concerning the choice of the probability density (pdf) of marginals for the estimation of copulas is done via kernel density estimation (KDE) [42]. The classifier is learned by fitting the multivariate distribution of the pre-processed image with Gaussian, t-Student, Clayton, Gumbel and Frank copulas, than, it is used to improve the prediction of the test set. Preliminary results of the proposed algorithm are presented in [43]. Here the proposed method has been tested on single hyperspectral images acquired by the ROSIS and AVIRIS sensors. Experimental results confirm that the use of copulas for the classification achieves equal or better accuracy in comparison with other modern methods here considered. An extension of this algorithm for time series based on the use of Bernstein copulas has been used in [44] for the classification of High-Resolution Satellite Image Time Series (SITS).

The rest of the paper is structured as follows; Section 2 introduces the mathematical theoretical tools used for HS classification, Section 3 describes the algorithm CopSCHI, Section 4 provides the details of the experiments, the results and some discussions about them. In particular, the experiments described show the effectiveness of the proposed methodology and compare its performance with various recent competitors. Finally, Section 5 summarizes the conclusions.

2. Mathematical foundations

2.1. Copulas

In statistical theory, copulas are the mechanism which allows to isolate the dependency structure in a multivariate distribution [32]. In particular, we can build any multivariate distribution by specifying the marginal distributions and the copula, separately. Although the copula functions have been used to model linear and nonlinear dependencies, they have been seldom used in computer science applications where nonlinear dependencies are common and need to be represented [45,46]. The complete treatment of the copulas was made in [32,33,47], here, we briefly recall only the fundamental concepts:

Definition 2.1. A copula C is a joint distribution function of standard uniform random variables. That is,

$$C(u_1,\dots,u_d) = P(U_1 \le u_1,\dots,U_d \le u_d),$$

where $U_i \sim U(0,1)$ for $i = 1,\dots,d$.

The following theorem constitutes the fundamental result in the context of copulas. This is known as the Sklar's theorem, and gives the relationship between a joint distribution and the relative cumulative function of copula.

Theorem 2.2 (*Sklar's Theorem*). *Let F be a d-dimensional distribution function with marginals $F_1, F_2, \dots, F_d$, then there exists a copula C such that for all x in $\overline{\mathbb{R}}^d$ with components $(x_1,\dots,x_d)$,*

$$F(x_1,x_2,\dots,x_d) = C(F_1(x_1), F_2(x_2),\dots,F_d(x_d)),$$

where $\overline{\mathbb{R}}$ denotes the extended real line $[-\infty,\infty]$. If $F_1(x_1), F_2(x_2),\dots,F_d(x_d)$ are all continuous, then C is unique. Moreover C is uniquely determined on $range(F_1) \times range(F_2) \times \cdots \times range(F_d)$.

According to Definition 2.1 and Theorem 2.2, any joint distribution function F with continuous marginals $F_1, F_2, \dots, F_d$ has associated a unique copula function C. Furthermore, the corresponding copula C is a function of the marginal distributions $F_1, F_2, \dots, F_d$. A very useful result of Theorem 2.2 is that the d-dimensional joint density f and the marginal densities $f_1, f_2, \dots, f_d$ are also related:

$$f(x_1,\dots,x_d) = c(F_1(x_1),\dots,F_d(x_d)) \times \prod_{i=1}^{d} f_i(x_i), \tag{1}$$

where c is the density of the copula C. Eq. (1) shows that the product of marginal densities and a copula density builds a d-dimensional joint density. Notice that the dependence structure is given by the copula function and the marginal densities can be of different distributions. This differs from the classical method to build multivariate distributions, where the main limitation is the assumption that the marginals should be generally of the same type. The flexibility of copula functions consists in the possibility to differentiate the marginal distribution from the probability joint structure. In this work we use the two main copula's families, Elliptical and Archimedean families and among them we employ the Gaussian, t-Student, Gumbel, Clayton and Frank copulas [34].

2.2. The probabilistic classifier based on copula function

In this section we introduce the classifier using copulas, following the strategy in [40]. Differently from the methodology used in [40], in our approach we do not investigate a single copula but dynamically choose the best one from a set of copulas. We recall that if F is an absolutely continuous multivariate distribution function with marginals $F_1,\dots,F_d$, than the *join* pdf f can be expressed by

$$f(x_1,\dots,x_d;\boldsymbol{\Phi},\boldsymbol{\Psi}) = c(F_1(x_1,\phi_1),\dots,F_d(x_d,\phi_d);\boldsymbol{\Psi}) \times \prod_{k=1}^{d} f_k(x_k;\phi_k) \tag{2}$$

where $c(u_1,\dots,u_d) = \frac{\partial^d C(u_1,\dots,u_d)}{\partial u_1 \dots \partial u_d}$ represents the density of the copula $C(u_1,\dots,u_d)$ and f_k denotes the pdf of F_k, $k = 1,\dots,d$, $\mathbf{\Phi} = \{\phi_k | k = 1,\dots,d\}$ represents the parameters of the marginals, $\mathbf{\Psi}$ are the parameters with respect to the copula (which may be a unique parameter or multiple parameters depending on the selected copula). Let $\Omega = \{\omega_1,\dots,\omega_m\}$ be a finite set of m classes and suppose we want to assign to each x from the space $\overline{\mathbb{R}^d}$ a class from Ω. By using the Bayesian decision theory [48], x can be assigned to the class ω_i if:

$$g_i(x) > g_j(x) \qquad \text{for all} \qquad j \neq i$$

where, for all i $g_i : [0,\infty)^d \to \mathbb{R}$ are called discriminant functions that are defined by

$$g_i(x) = \mathbf{P}(\omega_i|x) = \frac{f(x|\omega_i)\mathbf{P}(\omega_i)}{\sum_{j=1}^{m} f(x|\omega_j)\mathbf{P}(\omega_j)} \tag{3}$$

where $f : \mathbb{R}^d \to [0,\infty)$ is a likelihood function and $\mathbf{P}(\omega_i)$, $i = 1,\dots,m$ are the prior distributions of the classes from Ω.

Using (2), the likelihood $f(x|\omega_i)$ can be written as:

$$\begin{aligned} &f(x_1,\dots,x_d;\boldsymbol{\Phi},\boldsymbol{\Psi}|\omega_i) = \\ &\qquad c(F_1(x_1;\phi_1|\omega_i),\dots,F_d(x_d;\phi_d|\omega_i);\boldsymbol{\Psi}|\omega_i) \times \prod_{k=1}^{d} f_k(x_k;\phi_k|\omega_i). \end{aligned} \tag{4}$$

and then the discriminant function (3) for every i becomes,

$$\begin{aligned} g_i(x) &= \mathbf{P}(\omega_i|x) \\ &= \frac{\Big(c\big(F_1(x_1;\phi_1|\omega_i),\dots,F_d(x_d;\phi_d|\omega_i);\boldsymbol{\Psi}|\omega_i\big)\prod_{k=1}^{d} f_k(x_k;\phi_k|\omega_i)\Big)\mathbf{P}(\omega_i)}{\sum_{j=1}^{m}\Big(c\big(F_1(x_1;\phi_1|\omega_j),\dots,F_d(x_d;\phi_d|\omega_j);\boldsymbol{\Psi}|\omega_i\big)\prod_{k=1}^{d} f_k(x_k;\phi_k|\omega_j)\Big)\mathbf{P}(\omega_j)}. \end{aligned} \tag{5}$$

The equation above depends on the conditional copula density, the conditional marginal densities and the prior probability of the class.

2.3. Fitting copula

To estimate the conditional copula density parameters within Eq. (5), the well-known methods of copula theory can be used. These methods are generally divided into parametric and non-parametric ones. In the first case, assuming that there are some evidences that the marginals belong to known distributions, we can estimate the parameters $\mathbf{\Phi} = \{\phi_k | k = 1,\dots,d\}$ and $\mathbf{\Psi}$ by *maximum likelihood estimation* (MLE), i.e. the maximization of the likelihood function:

$$\begin{aligned} &\log(\mathcal{L}(\mathbf{\Phi},\mathbf{\Psi})) \\ &= \sum_{i=1}^{N}\log\Bigg(c(F_1(x_{i,1};\phi_1),\dots,F_d(x_{i,d};\phi_d),\mathbf{\Psi})\Bigg) + \Bigg(\sum_{i=1}^{N}\sum_{j=1}^{d}\log f_j(x_{i,j};\phi_j)\Bigg). \end{aligned} \tag{6}$$

A computationally advantageous technique for maximizing (6), called *inference for marginals* (IFM) was developed in [34]. This method is an estimation approach for a multivariate (non-normal) response with co-variates when each of the parameters (either a uni-variate or a dependence parameter) of the model can be associated with a marginal distribution [34,47]. The approach consists of estimating uni-variate parameters from separately maximizing uni-variate likelihoods, and then estimating copula dependence parameters from separate bi-variate likelihoods or from a multivariate likelihood. More specifically, the log-likelihoods in the second part of Eq. (6) of the d univariate marginals are separately maximized to get estimates of $\mathbf{\Phi} = \{\phi_k | k = 1,\dots,d\}$ and the first part of Eq. (6) is maximized to get the estimate of $\mathbf{\Psi}$.

In many real-world situations, it is not easy to specify marginal distributions, especially when the problem being analyzed involves large data. In this context, the non-parametric approach allows this problem to be addressed. This method leads to estimating only the copula parameter using *pseudo-observations*, i.e. the empirical cumulative distribution function $\hat{F}_j$, $j = 1\dots d$ of the marginals [49]. Moreover, also the marginals pdf f_j, $j = 1\dots d$ in (6) can be estimated non parametrically. In this work we adopted the useful Kernel Density Estimation (KDE) [42]. In more detail, let $x_1, x_2, \dots, x_k$, be k samples drawn from an unknown distribution. Then for any value x the formula for KDE is:

$$\hat{f}(x;h) = \frac{1}{kh}\sum_{i=1}^{k} K\left(\frac{x - x_i}{h}\right),$$

where $K(.)$ is the Gaussian kernel $K(x) = \frac{1}{\sqrt{2\pi}}exp\left(-\frac{x^2}{2}\right)$ and h is the bandwidth which controls the smoothness of the resulting density curve. Among the techniques suggested in [42] to estimation h, we have chosen the Improved Sheather Jones (ISJ) algorithm, as in the analysis of HS images, the sample dimension is big and far from being normal. The Improved Sheather-Jones algorithm is a plug-in selector. The mean integrated square error (MISE) is given by

$$MISE(h) = \mathbb{E}\int\left(\hat{f}(x;h) - f(x)\right)^2 dx.$$

The ISJ algorithm attempts to find h to minimize the asymptotic mean integrated square error (AMISE), which depends on the unknown quantity $\|f''(x)^2\|$. Using a recursive formula, this is accomplished by computing a sequence of estimates [50].

Once the empirical distribution functions $\hat{F}_j$ and $\hat{f}_j$ have been calculated, Eq. (6) can be written as:

$$\begin{aligned} &\log(\mathcal{L}(\mathbf{\Phi},\mathbf{\Psi})) \\ &= \sum_{i=1}^{N}\log\Bigg(c(\hat{F}_1(x_{i,1}),\dots,\hat{F}_d(x_{i,d}),\mathbf{\Psi})\Bigg) + \Bigg(\sum_{i=1}^{N}\sum_{j=1}^{d}\log\hat{f}_j(x_{i,j})\Bigg). \end{aligned} \tag{7}$$

In Eq. (7) the only parameter to be estimated is $\mathbf{\Psi}$.

$$\hat{\mathbf{\Psi}} = \underset{\mathbf{\Psi}}{\operatorname{argmax}}\,\mathcal{L}(\mathbf{\Psi}) = \underset{\mathbf{\Psi}}{\operatorname{argmax}}\sum_{i=1}^{N}\log\Bigg(c(\hat{F}_1(x_{i,1}),\dots,\hat{F}_d(x_{i,d}),\mathbf{\Psi})\Bigg). \tag{8}$$

This procedure is called *maximum pseudo-likelihood estimation* (MPLE) since the pseudo-observations are considered and it is proven to be computationally more advantageous than the parametric method [49].

2.4. Singular value decomposition

In general, remote sensing imagery consists of large arrays. The analysis of these matrices, in the task of classification, requires a large storage space and a very high computational cost; furthermore, the use of copulas is more efficient for analysis and inference of data of small size. Low-rank matrix dimensionality reduction process represents a branch of unsupervised mathematical techniques, devoted to either to extract or create a low-dimensional structure which preserves the most important information [51–53]. The SVD is one of the most powerful tools for decompose a matrix. It has several advantage: it always exists for any matrix, is numerically stable, is data-driven and can be used in different domains where the data can be reorganized in the form of a matrix.

Let $X \in \mathbb{R}^{n\times d}$ denote an $n \times d$ matrix of real-valued data with rank r, where without loss of generality $n \geq d$, and therefore $r \leq n$. The singular value decomposition is given by:

$$X = UDV^T \tag{9}$$

where U is an *orthogonal* matrix of dimension $n \times n$, D is a rectangular matrix of dimension $n \times d$, and V^T is a orthogonal square matrix of dimension $d \times d$ [53]. When n is greater than d, the matrix D may be written as:

$$D = \begin{bmatrix} \Sigma \\ 0 \end{bmatrix}, \quad \text{with} \quad \Sigma = diag(\sigma_1, \dots, \sigma_d).$$

The elements on the diagonal of Σ are non-negative and arranged in non-increasing order.

The *truncated* form of SVD is used to represent X, that is:

$$X = UDV^T = \begin{bmatrix} U_d & {U_d}^{\perp} \end{bmatrix} \begin{bmatrix} \Sigma \\ 0 \end{bmatrix} V^T = U_d \Sigma V^T. \tag{10}$$

The matrix U_d contains the first d principal columns of U. The matrix ${U_d}^{\perp}$ contains the columns that generates an orthogonal vector space complementary to the one generated by U_d. The diagonal elements of the matrix Σ, $\sigma_1, \dots, \sigma_d$ are called singular values, instead the columns of U and V are called left and right singular vectors, respectively [53]. The number of non-zero singular values is the rank of X. The SVD low-rank approximation of X is obtained considering the principal r singular values, moreover the Schmidt's approximation theorem [54] states that the optimal rank-r approximation to X, in a least squares sense, is given by the rank-r SVD truncation $U_r \Sigma_r {V_r}^T$, that corresponds to a sum of rank-1 matrices:

$$\underset{\tilde{X}\ \text{s.t.}\ rank(\tilde{X})=r}{\operatorname{argmin}} \|X - \tilde{X}\|_p = U_r \Sigma_r {V_r}^T = \sum_{i=1}^{r} \sigma_i u_i v_i^T.$$

Here, Σ_r contains the principal $r \times r$ sub-block of Σ; $\| \cdot \|_p$ is either the 2-norm or the Frobenius norm. and u_i, v_i are the columns of the matrices U, V.

The low rank matrix factorization strategy described above allows us to build a low-dimension approximation of the images in terms of most significant spectral features.

3. Classification algorithm CopSCHI

In this paper we develop a new method for the classification called CopSCHI (Copula Supervised Classification of Hyperspectral Images).

As known in machine learning theory, the problem of classification consists in learning some unknown function that maps an input x (which could be a vector) into an output y where $y \in \{1, \dots, C\}$, with C being the number of classes. This function is estimated on the basis of labeled training data consisting of a set of training examples. The classification algorithm analyzes training data and generates a deducted function, which can be used to make predictions on novel inputs, meaning ones that we have not seen before. Therefore, in the classification task, the first fundamental step is to decide what type of data should be used as a training set. Thus, a set of input objects is collected and the corresponding outputs are also collected, both by human experts and by measurements.

In our case the aim is to classify images that come from satellite hyperspectral sensors. One way to formalize the problem is to denote with $\mathcal{I}$ the HS image, $\mathcal{I}$ can be seen as a tensor of dimension $n \times m \times d$ where, n and m identify the position of the pixels and d represents the number of spectral signatures. The image (tensor) $\mathcal{I}$ can be reorganized in a matrix I of dimension $p \times d$ where $p = n \times m$. Often for this type of images there is no information of ground truth (labels) for each pixel, therefore it is necessary to select as training set only that subset of pixels for which we have information.

The algorithm CopSCHI we propose consists in the following phases:

Dimensionality reduction phase:

1. Application of the truncated SVD algorithm (10) to reduce the dimensionality of the training set, previously reorganized as a matrix X, by selecting an appropriate number of singular values $\tilde{r}$. Deciding how many singular values to keep, i.e. where to truncate, is one of the most important and contentious decisions when using the SVD [53].
   There are many factors, including specifications on the desired rank of the system, the magnitude of noise, and the distribution of the singular values. It is a common praxis to truncate the SVD at a rank $\tilde{r}$ that captures a pre-determined amount of the

variance or energy in the original data, such as 90% or 99% truncation. The amount of overall variance explained by the i-th pair of SVD vectors is given by:

$$R_i^2 = \frac{\sigma_i^2}{\sum_j \sigma_j^2}.$$

This can also be computed as the ratio of the Frobenius norm of the rank-1 reconstructions to the norm of the original data matrix:

$$R_i^2 = \frac{\|\sigma_i u_i v_i^T\|_F^2}{\|X\|_F^2} = \frac{\sigma_i^2}{\sum_j \sigma_j^2},$$

where u_i and v_i are i-th columns of U and V correspondingly. It is also possible to use the ratio of the 2-norm of rank-1 reconstruction to the 2-norm of the original data matrix:

$$E_i = \frac{\|\sigma_{i+1} u_{i+1} v_{i+1}^T\|_2}{\|X\|_2} = \frac{\sigma_{i+1}}{\sigma_1}.$$

We observe that $E_i = \|X - U_i \Sigma_i V_i^T\|_2 / \|X\|_2, i = 1, n$ is the relative error in the approximation of the original matrix using the 2-norm, so it gives information about the quality of the approximation. The rank one matrices with R_i^2 or E_i larger than a threshold τ are kept, while the remaining matrices are truncated.
Once $\tilde{r}$ has been chosen, we get the reduced image of the training set $I^{\tilde{r}} = U_{\tilde{r}} \Sigma_{\tilde{r}}$.

Learning phase:

2. The pixels of the training dataset are grouped by class using the representation given by $I^{\tilde{r}}$ and the labels vector. For each class the marginals are evaluated empirically and subsequently, for each class the copula is automatically fitted to the data dynamically choosing among Gaussian, Frank, Gumbel and Clayton copulas belonging to Elliptical and Archimedean families. To estimate the copula and find its parameter, the log-likelihood function (8) and the MPLE procedure have been adopted as described in Section 2.3. The best copula representing the class is selected based on the minimum value of the Akaike Information Criterion (AIC). For the maximization of the log-likelihood we use a numerical method, called Limited-Memory BFGS (L-BFGS or LM-BFGS) that is an optimization algorithm in the family of quasi-Newton methods that approximates the Broyden–Fletcher–Goldfarb–Shanno algorithm (BFGS) using a limited amount of computer memory. It is a popular algorithm for parameter estimation in machine learning [55,56]. The advantage of L-BFGS is that it requires a smaller number of gradients instead of $\frac{n(n+1)}{2}$ elements needed to store the whole (triangle) of a Hessian estimate, as is required with BFGS, where n is the size of the problem. Unlike BFGS, the Hessian estimation is never explicitly formed or stored in L-BFGS, rather, the calculations that would be necessary with the Hessian estimation are done without explicitly forming it. L-BFGS is used instead of BFGS when n is very large.

Predicting phase:

3. In this phase, we select data (reorganized in the matrix I) for which we have no information to predict the class learned by the copula classifier CopSCHI.
Since we have applied dimensionality reduction in the training phase, for the observation represented by t (a row of the matrix I) we evaluate the projection $y = tV_{\tilde{r}}^T$, that is a vector of dimension $1 \times \tilde{r}$. Then, taking into account the density of the copula c previously fitted in the learning phase on the training set, for each of the m classes the discriminant functions $g_i(y)$, $i = 1 \dots m$, are calculated on y by using (5) in order to obtain the probabilities given the observation that it belongs to the specific class. Finally, we get a prediction vector that assigns the pixel to the class for which it has the greatest probability calculated with the discriminant function.

In the next section we describe the experiments conducted and the details for the initialization of the CopSCHI algorithm.

4. Experiments

In order to demonstrate the effectiveness of our algorithm we will consider two datasets that consist both in a single HS remote sensing scene and we will compare the results with benchmark machine learning algorithms and advanced classification techniques based on neural networks architectures. More in detail, we will consider, Random Forest (RF) [5], Support Vector Machine (SVM) [57], Long Short Time Memory (LSTM) [58] and Convolutional Neural Network (CNN) [27]. The first two methods have been chosen because they still represent a fast and stable approach for classification analysis, therefore are widely used for multi-class land cover classification and with high-dimensional data. In addition, to demonstrate the validity of the use of copulas in the classification task, we also take into consideration deep learning algorithms which, over the past decade, have been used to a greater extent for RS images classification. In the literature different NN architectures has been developed, especially by employing CNN that are suitable for the classification of images. A complete comparison with all these architectures of NN is beyond the scope of this work, for this reason in this paper we consider, as competitors, only two techniques that involve the CNN and LSTM. The accuracy performance is evaluated in term of Overall Accuracy (OA), F1-score and K measure [29,59,60]. The use of K measure is justified by the fact that this index is widely used especially in the presence of classifications in which the number of labeled observations within the classes are unbalanced. The presentation of the results is organized as follows: description of the Dataset, experimental settings and Numerical results.

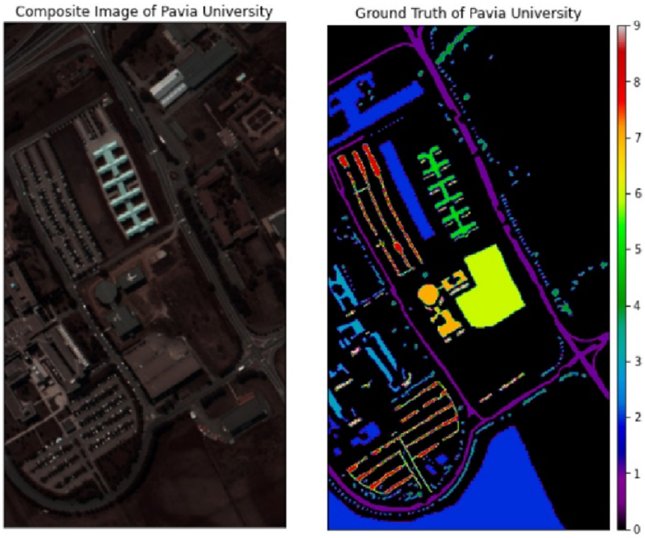

Composite Image of Pavia University
Ground Truth of Pavia University
9
8
7
6
5
4
3
2
1
0

Fig. 1. RGB Composite of Pavia University (*left*) and Ground Truth of Pavia University with different colors for each category(*right*). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

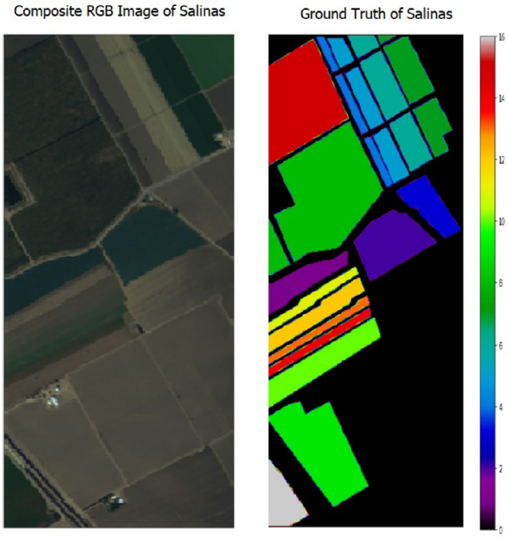


Fig. 2. RGB Composite of Salinas (*left*) and Ground Truth of Salinas with different colors for each category(*right*). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4.1. Dataset

We have considered two Hyperspectral remote sensing scenes widely used for testing algorithms.[1] The first dataset is called the Pavia University dataset. It is an urban site over the University of Pavia, Italy. The dataset was captured by the reflective optics system imaging spectrometer (ROSIS-3). The size of the image is 610 × 340 with 1.3 m spatial resolution. The image has 103 spectral bands prior to water-band removal. It has a spectral coverage of $0.43 - 0.86$ μm. A false composite image (R-G-B = band 10-27-46) and the corresponding ground truth are shown in Fig. 1. The second dataset is called Salinas. This scene was collected by the 224-band AVIRIS sensor over Salinas Valley, California, and is characterized by high spatial resolution (3.7-meter pixels). The area covered comprises 512 lines by 217 samples. We discarded the 20 water absorption bands, in this case bands: (108–112), (154–167), 224. This image was available only as at-sensor radiance data. It includes vegetables, bare soils, and vineyard fields. Salinas ground truth contains 16 classes, Fig. 2.

[1] http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes.

Table 1
Quantitative per class analysis, F1-score the Pavia University dataset.

Method	1-Asphalt	2-Meadows	3-Gravel	4-Trees	5-Painted metal sheets	6-Bare Soil	7-Bitumen	8-Self Blocking Bricks	9-Shadows
RF	90.18%	91.63%	57.21%	90.58%	99.87%	62.67%	71.41%	81.25%	99.64%
SVM	98.01%	96.12%	68.21%	72.88%	96.87%	61.29%	54.11%	83.91%	99.96%
LSTM	95.17%	95.44%	67.12%	75.98%	98.09%	63.73%	68.14%	80.37%	99.91%
CNN	93.02%	95.10%	71.47%	77.02%	98.66%	64.31%	70.29%	85.04%	100.00%
CopSCHI	94.33%	96.18%	81.71%	90.10%	99.63%	90.60%	91.10%	88.14%	100.00%

Table 2
Quantitative per class analysis, F1-score the Salinas dataset.

Method	1-Brocoli green weeds 1	2-Brocoli green weeds 2	3-Fallow	4-Fallow rough plow	5-Fallow smooth	6-Stubblet	7-Celery	8-Grapes untrained	9-Soil vinyard develop
RF	99.50%	99.64%	97.27%	99.28%	98.38%	99.92%	99.67%	82.16%	99.36%
SVM	97.18%	99.06%	84.98%	98.36%	94.89%	99.69%	98.78%	76.51%	99.32%
LSTM	95.38%	99.09%	88.97%	97.41%	98.77%	99.14%	99.01%	77.41%	98.16%
CNN	99.11%	99.07%	96.78%	96.99%	98.04%	99.90%	99.69%	84.79%	99.10%
CopSCHI	99.75%	99.73%	98.11%	99.64%	97.95%	99.92%	99.72%	83.74%	99.16%
Method	10-Corn senesced green weeds	11-Lettuce romaine 4wk	12-Lettuce romaine 5wk	13-Lettuce romaine 6wk	14-Lettuce romaine 7wk	15-Vinyard untrained	16-Vinyard vertical trellis		
RF	95.70%	95.56%	97.8%	97.47%	96.51%	63.45%	99.17%		
SVM	82.03%	70.42%	93.03%	92.70%	90.28%	57.67%	95.39%		
LSTM	86.19%	82.34%	88.92%	90.72%	85.07%	58.24%	94.10%		
CNN	87.05%	83.18%	96.91%	94.88%	90.16%	60.47%	97.91%		
CopSCHI	94.34%	98.75%	99.22%	99.27%	97.18%	75.92%	99.26%		

4.2. Experimental settings

CopSCHI is written in Python 3.7. Experiments are launched using an Intel(R) Core(TM) i7-9750H CPU @ 2.60 GHz and 16 GB RAM running Microsoft Windows 8.1 (64 bits).

For our analysis, as preliminary phase for our algorithm described in Section 3, the dataset has been split into $k\%$ of training set, the rest $(100-k)\%$ of the dataset has been used in the predicting phase as testing set. The training set has been standardized, so that the mean of observed values is 0 and the standard deviation is 1. The choice of splitting size percentage k can vary, first we have selected $k = 50$. The training set has been used as input for our algorithm, and for the initialization of the SVD we have selected a number of components $\tilde{r} = 20$ for Pavia University and $\tilde{r} = 22$ for Salinas. This choice of $\tilde{r}$ is made by considering the amount of overall variance explained, setting an high value of variance, 0.995% for Pavia dataset and 0.99% for Salinas dataset, as explained in Section 3. It is worth noting that we have split the dataset only in a training and test set without taking into account a validation set. This because, in the experiments carried out in this work we do not use a validation set but obviously this can be defined in the preliminary phase in order to tune the parameters.

In the learning phase, we select for each copula a random initial parameter and we employ the optimization algorithm L-BFGS in order to find the best parameter. When the choice of copula is enabled, the best is selected with respect to the minimum value of AIC otherwise when only one copula is chosen then the best parameter is selected according to the maximum log-likelihood function. For the competitors, we have implemented RF by adopting the default parameter set-up reported in the documentation.[2] For SVM, we have performed a cross validation[3] in order to estimate the parameter and we have selected as kernel the radial basis function (RBF). For the experiments conducted by using NN architectures, LSTM and CNN we have used the same parameters described in [16] using the Python TensorFlow library.[4]

4.3. Numerical results

To evaluate the performance of the CopSCHI algorithm, we considered a test set together with the vector of the observed labels. We have compared the predicted vectors of labels, obtained with our approach, with the other classifiers chosen as competitors. The quantitative analysis was performed by using the metrics cited in Section 4. Tables 1 and 2, depict the average results in terms of F1-score reached by the competitors classifiers and the best result obtained with CopSCHI setting the configuration in which we choose different copula functions amongst the two families, Euclidean and Archimedean.

Even if it is always not easy to compare different algorithms, the results show that the proposed classifier outperforms the chosen competitors. In particular, for Pavia University dataset, we can observe that, considering the main competing approaches

[2] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html.

[3] https://scikit-learn.org/stable/auto_examples/model_selection/plot_grid_search_digits.html.

[4] https://www.tensorflow.org/.

Table 3
Percentage in term of F1-score for each classes in Pavia University dataset with CopSCHI with different copulas and one copulas. Best copula functions estimated for each class (column 4), respective AIC value (column 5) and F1-score (column 3). Results with only Gaussian copula in term of F1-score (column 6) and relative AIC (column 8).

Class	Support	F1-score	Different copulas	AIC	F1-score	Gaussian copula	AIC
Asphalt	1989.00	94.91%	Gaussian	−36 650	94.91%	Gaussian	−79 279
Meadows	5595.00	96.46%	Gaussian	−79 279	96.46%	Gaussian	−79 279
Gravel	630.00	81.74%	Clayton	−14 827	79.34%	Gaussian	−13 827
Trees	919.00	91.95%	Clayton	−13 550	88.74%	Gaussian	−2382
Painted metal sheets	403.00	100.00%	Gumbel	−9091	98.54%	Gaussian	−1827
Bare Soil	1509.00	90.38%	Gaussian	−21 970	90.38%	Gaussian	−21 970
Bitumen	399.00	91.73%	Gaussian	−8909	91.73%	Gaussian	−8909
Self-Blocking Bricks	1105.00	88.28%	Gaussian	−18 712	88.28%	Gaussian	−18 712
Shadows	284.00	100.00%	Frank	−5652	100.00%	Gaussian	−15 652

Table 4
Percentage in term of F1-score for each classes in Salinas University dataset with CopSCHI with different copulas and one copulas. Best copula functions estimated for each class (column 4), respective AIC value (column 5) and F1-score (column 3). Results with only Gaussian copula in term of F1-score (column 6) and relative AIC (column 8).

Class	Support	F1-score	Different copulas	AIC	F1-score	Gaussian copula	AIC
Brocoli green weeds 1	603.00	99.75%	Gaussian	−24 830.0	99.75%	Gaussian	−24 830.0
Brocoli green weeds 2	1118.00	99.73%	Gaussian	−36 940.3	99.73%	Gaussian	−36 940.3
Fallow	593.00	98.11%	Frank	−28 620.2	96.11%	Gaussian	−23 109.2
Fallow rough plow	418.00	99.64%	Clayton	−23 125.0	98.64%	Gaussian	−21 436.0
Fallow smooth	803.00	97.95%	Gaussian	−35 144.4	97.95%	Gaussian	−35 144.4
Stubblet	1188.00	99.92%	Gaussian	−39 599.1	99.92%	Gaussian	−39 599.1
Celery	1074.00	99.72%	Gaussian	−48 125.9	99.72%	Gaussian	−48 125.9
Grapes untrained	3381.00	83.74%	Gaussian	−78 175.1	83.74%	Gaussian	−78 175.1
Soil vinyard develop	1861.00	99.16%	Gaussian	−65 467.3	99.16%	Gaussian	−65 467.3
Corn senesced green weeds	983.00	94.34%	Gumbel	−27 943.2	93.77%	Gaussian	−24 693.2
Lettuce romaine 4wk	320.00	98.75%	Gumbel	−12 236.0	97.82%	Gaussian	−11 561.8
Lettuce romaine 5wk	578.00	99.22%	Gumbel	−21 045.1	98.89%	Gaussian	−13 086.0
Lettuce romaine 6wk	275.00	99.27%	Gaussian	−9507.7	99.27%	Gaussian	−9507.7
Lettuce romaine 7wk	321.00	97.18%	Frank	−8685.3	95.04%	Gaussian	−6742.3
Vinyard untrained	2181.00	75.92%	Gaussian	−55 565.6	75.92%	Gaussian	−55 565.6
Vinyard vertical trellis	542.00	99.26%	Gaussian	−25 753.7	99.26%	Gaussian	−25 753.7

our framework supplies the best classification results on seven over nine land cover classes. These classes are: 2-Meadows, 3-Gravel, 5-Painted metal sheets, 6-Bare Soil, 7- Bitumen, 8- Self Blocking Bricks and 9-Shadows. For Salinas our methodology achieves the best result on thirteen over sixteen land cover classes. This classes are: 1-Broccoli green weeds1, 3- Fallow, 4-Fallow rough plow, 7-Celery, 8-Grapes untrained, 10-Corn senesced green weeds, 11-Lettuce romaine 4wk, 12-Lettuce romaine 5wk, 13-Lettuce romaine 6wk, 14-Lettuce romaine 7wk, 15-Vineyard untrained and 16-Vineyard vertical trellis.

In Tables 3, 4 are reported the results obtained by launching the algorithm first, with the automatic copula selection and then comparing them with a configuration in which only the Gaussian copula is considered.

Looking at the results in terms of AIC and F1-score it can be observed how the choice of copulas improves the result of class classification, this justifies the use of this tool which allows to have different copulas able to model multivariate distributions. We want to address the fact that in this work, only five copula functions have been considered and that many others exist in the literature [47]. We also noticed that the choice of the t-Student copula greatly slowed down the execution of the algorithm and therefore we decided to exclude it from the choice also considering the fact that the results obtained did not change. Finally we observe, without reporting the results, that in addition to testing the single Gaussian copula we also tested all the others types individually. The results obtained in terms of accuracy decrease, this is justified by the fact that in general, when the size of the dataset is very big, the Gaussian copula is a better fit to the data [32]. In fact, in the two datasets taken into consideration, looking at the column relating to the different copulas only four out of nine classes for the Pavia dataset are fitted with Archimedean copulas, while for Salinas only six out of sixteen. In addition to show the robustness of the proposed approach, for all the algorithms considered we have performed a Stratified K-Fold Cross-Validation[5] with $k = 5$. The average results in terms of Accuracy, F1-score and K measure are reported in Tables 5 and 6 and we can observe that CopSCHI still outperforms all the competing methods.

Moreover, it has the smallest variance in all the considered measures. We observe also that the variance of the K measure is really small, thus showing that it is not so sensible to this parameter, that is considered the best one when the classes are not balanced. It is worth noting that, using different percentage of the split of the data, the accuracy of the different classifier may vary, as simpler or more difficult examples are involved in the training or test set. For this reason, in Figs. 3 and 4, in order to confirm the stability of our method, we report the results, in terms of Accuracy, F1-score and K measure [59], by varying the percentage of the training dataset.

[5] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedKFold.html#sklearn.model_selection.StratifiedKFold.

Table 5
Comparison analysis for the classification of Pavia University dataset performing a K-fold Cross Validation with $k = 5$.

	RF	SVM	LSTM	CNN	CopSCHI
Accuracy	88.25% ± 0.25%	82.77% ± 1.33%	84.37% ± 1.83%	84.12% ± 2.46%	92.93% ± 0.49%
F1-score	87.12% ± 1.48%	81.89% ± 1.06%	84.01% ± 2.58%	83.36% ± 1.67%	92.94% ± 0.28%
K	83.07% ± 0.59%	76.54% ± 1.32%	77.82% ± 2.43%	80.12% ± 2.08%	90.65% ± 0.25%

Table 6
Comparison analysis for the classification of Salinas University dataset performing a K-fold Cross Validation with $k = 5$.

	RF	SVM	LSTM	CNN	CopSCHI
Accuracy	87.90% ± 1.24%	84.07% ± 1.39%	84.48% ± 2.41%	86.03% ± 1.84%	92.87% ± 0.42%
F1-score	87.45% ± 0.35%	83.87% ± 1.68%	84.64% ± 0.94%	85.38% ± 2.47%	92.58% ± 0.17%
K	83.53% ± 0.84%	77.01% ± 1.47%	81.99% ± 2.09%	84.74% ± 3.45%	91.41% ± 0.19%

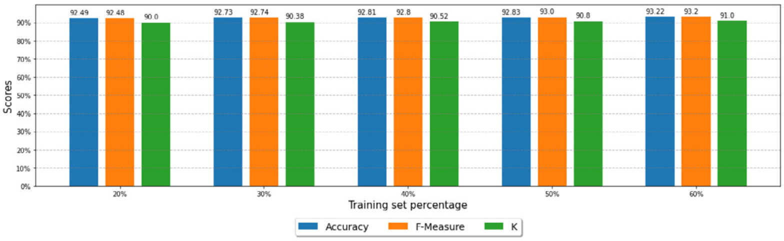


Fig. 3. Accuracy Measures, of CopSCHI, in term of OA, F1 Score and K for Pavia University by different percentages of training set.

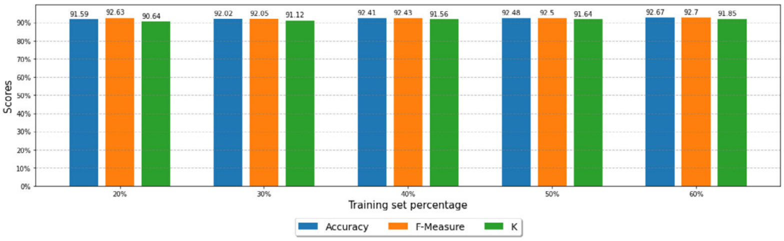


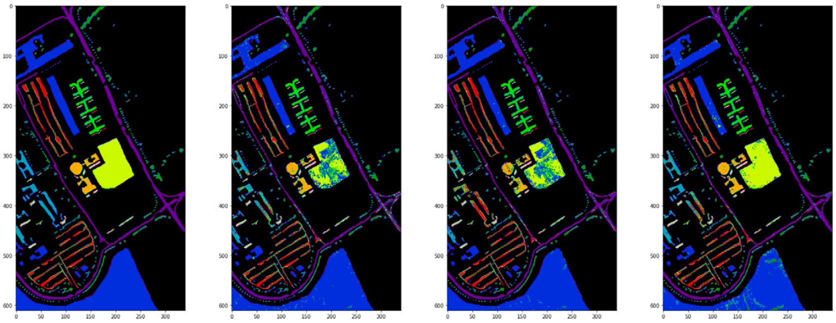
Fig. 4. Accuracy Measures, of CopSCHI, in term of OA, F1 Score, K for Salinas by different percentages of training set.

In particular the experiment has been conducted with a percentage of 20%, 30%, 40%, 50% and 60%, respectively. As can be seen in the barplots, for Pavia University the average, above the different percentages of the training set, is $92.81\% \pm 0.05\%$ for the Accuracy, $92.84\% \pm 0.06\%$ for F1-score and $90.54\% \pm 0.12\%$ for K. Instead, for Salinas dataset, we have an average of $92.23\% \pm 0.15\%$ for the Accuracy, $92.46\% \pm 0.05\%$ for F1-score and $91.36\% \pm 0.18\%$ for K measure. These results suggest to conclude that, even with low percentage of data for the training set, for both considered datasets, CopSCHI supplies high accuracy without high variance by varying the size of learning data. To conclude the analysis, in Figs. 5 and 6 we provide a qualitative investigation considering land cover maps produced by our approach versus those produced by two of four competitor methods. In summary, the qualitative analysis of the land cover maps produced for the Pavia University and Salinas study dataset confirms the quantitative results discussed in the Tables 1 and 2.

By making a careful visual analysis of the land cover maps of the various classifiers, it can be seen that CopSCHI provides the one that is very close to the ground truth. In fact, looking at Fig. 5(b) and (c) and comparing the maps with the ground truth in Fig. 5(a) it can be seen how the central part, corresponding to the bare soil class, appears non-uniform, with the presence of noise and therefore with pixels that are not correctly classified.

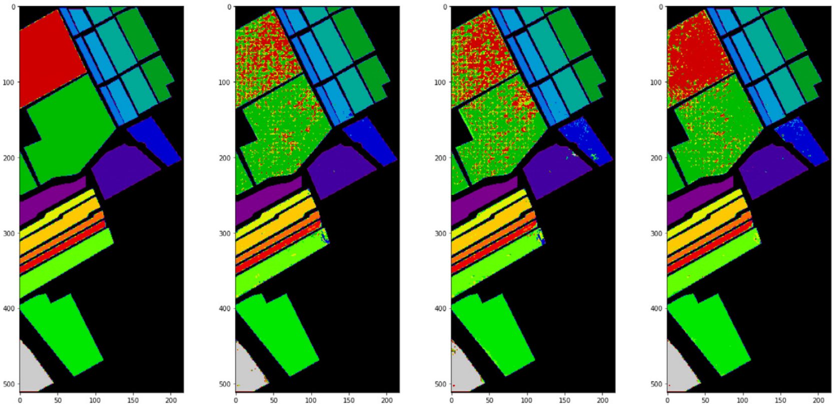
This type of noise is greatly reduced in the resulting land cover map produced by our method Fig. 5(d). Furthermore, regarding Fig. 6(b) and (c), it can be seen the red and green *salt and pepper* effect that represents misclassified pixel within the classes. This effect is less visible on the classification map Fig. 6(d) produced by our algorithm, confirming the quantitative results of CopSCHI.

In this direction, we want to highlight that, no post-processing stage was carried out downstream of the classification algorithm to eliminate that type of errors. Obviously such a correction could further improve the quantitative results reported in the previous Tables by CopSCHI.



(a) Ground Truth (b) RF (c) SVM (d) CopSCHI

Fig. 5. Qualitative investigation of Land Cover Map produced on the Pavia University study site: (a) Ground Truth, (b) RF, (c) SVM, and (d) CopSCHI.



(a) Ground Truth (b) RF (c) SVM (d) CopSCHI

Fig. 6. Qualitative investigation of Land Cover Map produced on the Salinas study site: (a) Ground Truth, (b) RF, (c) SVM, and (d) CopSCHI. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

5. Conclusion

In this paper we presented a new supervised learning framework consisting of two procedures combined, reduction of dimensionality through factorization of matrices and classification through the use of copula functions belonging to different families, in particular the family of Elliptic copulas and the family of Archimedean copulas. The choice of using copulas was guided by the fact that this tool is able to model the joint probability of random variables which, as often happens in real life, do not follow a Normal distribution. Once the factorization of the matrix through SVD has been applied and the number of singular value has been appropriately chosen, we obtain a representation of the HS image in a low dimension. Then in the learning phase, for each class of the reduced image, a fitting stage above different copula functions has been applied in order to choose the best probability representation of the class. This was possible thanks to an algorithm that automatically chooses the copula on the basis of the best AIC value. Experiments were also performed using a single copula, but it could be seen that the dynamic choice increased the accuracy of the classification. The proposed method achieved excellent results on two reference datasets, the Pavia University and Salinas data set, with an overall accuracy of 93,56%, 92,58% respectively. The main advantage of our framework is that it can be generalized to other remote sensing systems problems due to its robust design and the possibility to parallelize the entire process to obtain a final classification with a reduced computational cost. An important consideration is that the performance remains high despite varying the number of samples being trained. It is worth noting that, in the literature, various neural network architectures have been studied in the application of HS images classification with the dataset considered in this paper. The accuracy reached by this approaches is around 99% [61,62]. However, for all these methodologies some problems still remain open, first, the phase of training neural networks which requires large amounts of computational time, and second, these networks must be trained differently for each dataset to achieve high accuracy. The advantage of using our approach lies in the fact that it can be applied indiscriminately to any type of HS images, and it is less sensitive to the size of training set in the learning phase, this allows to obtain a high level of accuracy in a relatively short time. However, it has been observed that the performance in terms of OA is sensitive to the parameter chosen during the empirical estimation of the marginals. Moreover, in this work we use only five copula functions in order to fit the probability distribution of the classes in the learning stage. As future direction, a possible extension could be the automatic choice of parameters during the marginal estimation phase, in this context it could also be useful to use other techniques to fit the marginal distributions, like the techniques based on Spline Hermite quasi-interpolation presented in [63]. Another possible extension could be to insert additional copula functions to those used in this work.

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

We thank Antonella Falini and Joaquin Francisco Varela Bacur for their careful reading and for the constructive comments.
We thank the anonymous reviewers for their many insightful comments and suggestions which greatly improved the manuscript.
The authors are members of the INdAM Research group GNCS.

References

[1] Bégué A, Arvor D, Bellon B, Betbeder J, De Abelleyra D, P. D. Ferraz R, Lebourgeois V, Lelong C, Simões M, R. Verón S. Remote sensing and cropping practices: A review. Remote Sens 2018;10(1). http://dx.doi.org/10.3390/rs10010099.

[2] Guttler F, Ienco D, Nin J, Teisseire M, Poncelet P. A graph-based approach to detect spatiotemporal dynamics in satellite image time series. ISPRS J Photogramm Remote Sens 2017;130:92–107. http://dx.doi.org/10.1016/j.isprsjprs.2017.05.013.

[3] Khiali L, Ienco D, Teisseire M. Object-oriented satellite image time series analysis using a graph-based representation. Ecol Inform 2018;43:52–64. http://dx.doi.org/10.1016/j.ecoinf.2017.11.003.

[4] Dudani S. The distance-weighted k-nearest-neighbor rule. Syst, Man Cybern, IEEE Trans 1976;SMC-6:325–7. http://dx.doi.org/10.1109/TSMC.1976.5408784.

[5] Breiman L. Random forests. Mach Learn 2001;45(1):5–32.

[6] Cortes C, Vapnik VN. Support-vector networks. Mach Learn 2004;20:273–97. http://dx.doi.org/10.1007/BF00994018.

[7] Du P, Xue Z, Li J, Plaza A. Learning discriminative sparse representations for hyperspectral image classification. IEEE J Sel Top Sign Proces 2015;9(6):1089–104. http://dx.doi.org/10.1109/JSTSP.2015.2423260.

[8] Altalib G, Ahmed E. Land cover classification using hidden Markov models. Int J Comput Netw Commun Secur 2013;1(4):165–72.

[9] Samat A, Du P, Liu S, Li J, Cheng L. E^2 lms : Ensemble extreme learning machines for hyperspectral image classification. IEEE J Sel Top Appl Earth Obs Remote Sens 2014;7(4):1060–9. http://dx.doi.org/10.1109/JSTARS.2014.2301775.

[10] Li S, Song W, Fang L, Chen Y, Ghamisi P, Benediktsson J. Deep learning for hyperspectral image classification: An overview. IEEE Trans Geosci Remote Sens 2019;PP:1–20. http://dx.doi.org/10.1109/TGRS.2019.2907932.

[11] Datta D, Mallick P, Bhoi AK, Ijaz MF, Shafi J, Choi J. Hyperspectral image classification: Potentials, challenges, and future directions. Comput Intell Neurosci 2022;2022. http://dx.doi.org/10.1155/2022/3854635.

[12] Kakhani N, Mokhtarzade M, Zoej M. Deep learning spatial-spectral classification of remote sensing images by applying morphology-based differential extinction profile (DEP). Electronics 2021;10:2893. http://dx.doi.org/10.3390/electronics10232893.

[13] Ahmad M, Shabbir S, Roy SK, Hong D, Wu X, Yao J, Khan AM, Mazzara M, Distefano S, Chanussot J. Hyperspectral image classification—Traditional to deep models: A survey for future prospects. IEEE J Sel Top Appl Earth Obs Remote Sens 2022;15:968–99. http://dx.doi.org/10.1109/JSTARS.2021.3133021.

[14] Cheng G, Han J, Lu X. Remote sensing image scene classification: Benchmark and state of the art. Proc IEEE 2017;105:1865–83. http://dx.doi.org/10.1109/JPROC.2017.2675998.

[15] Chen Y, Lin Z, Zhao X, Wang G, Gu Y. Deep learning-based classification of hyperspectral data. Sel Top Appl Earth Observ Remote Sens, IEEE J 2014;7:2094–107. http://dx.doi.org/10.1109/JSTARS.2014.2329330.

[16] Ma A, Filippi AM, Wang Z, Yin Z. Hyperspectral image classification using similarity measurements-based deep recurrent neural networks. Remote Sens 2019;11(2). http://dx.doi.org/10.3390/rs11020194.

[17] Tanwar S, Ramani T, Tyagi S. Dimensionality reduction using PCA and SVD in big data: A comparative case study. 2018, p. 116–25. http://dx.doi.org/10.1007/978-3-319-73712-6_12.

[18] Makantasis K, Karantzalos K, Doulamis A, Doulamis N. Deep supervised learning for hyperspectral data classification through convolutional neural networks. In: 2015 IEEE international geoscience and remote sensing symposium (IGARSS). 2015, p. 4959–62. http://dx.doi.org/10.1109/IGARSS.2015.7326945.

[19] Herries G, Selige T, Danaher S. Singular value decomposition in applied remote sensing. 1996, p. 5/1 – 5/6. http://dx.doi.org/10.1049/ic:19960159.

[20] Jayaprakash C, Damodaran BB, V. S, Soman K. Dimensionality reduction of hyperspectral images for classification using randomized independent component analysis. In: 2018 5th International conference on signal processing and integrated networks (SPIN). 2018, p. 492–6. http://dx.doi.org/10.1109/SPIN.2018.8474266.

[21] Falini A, Castellano G, Tamborrino C, Mazzia F, Mininni RM, Appice A, Malerba D. Saliency detection for hyperspectral images via sparse-non negative-matrix-factorization and novel distance measures. In: 2020 IEEE conference on evolving and adaptive intelligent systems, EAIS 2020. IEEE; 2020, p. 1–8. http://dx.doi.org/10.1109/EAIS48028.2020.9122749.

[22] Appice A, Lomuscio F, Falini A, Tamborrino C, Mazzia F, Malerba D. Saliency detection in hyperspectral images using autoencoder-based data reconstruction. In: Helic D, Leitner G, Stettinger M, Felfernig A, Ras ZW, editors. Foundations of intelligent systems - 25th international symposium, ISMIS 2020, Graz, Austria, September 23-25, 2020, proceedings, vol. 12117. Lecture notes in computer science, Springer; 2020, p. 161–70. http://dx.doi.org/10.1007/978-3-030-59491-6_15.

[23] Chen Y, Zhao X, Jia X. Spectral–spatial classification of hyperspectral data based on deep belief network. IEEE J Sel Top Appl Earth Obs Remote Sens 2015;8(6):2381–92. http://dx.doi.org/10.1109/JSTARS.2015.2388577.

[24] Zhang L, Zhang L, Du B. Deep learning for remote sensing data: A technical tutorial on the state of the art. IEEE Geosci Remote Sens Mag 2016;4(2):22–40. http://dx.doi.org/10.1109/MGRS.2016.2540798.

[25] Bengio Y, Courville A, Vincent P. Representation learning: A review and new perspectives. IEEE Trans Pattern Anal Mach Intell 2013;35:1798–828. http://dx.doi.org/10.1109/TPAMI.2013.50.

[26] Xing C, Ma L, Yang X. Stacked denoise autoencoder based feature extraction and classification for hyperspectral images. J Sens 2015;2016:1–10. http://dx.doi.org/10.1155/2016/3632943.

[27] Hu W, Huang Y, Wei L, Zhang F, Li H. Deep convolutional neural networks for hyperspectral image classification. J Sens 2015;2015:1–12. http://dx.doi.org/10.1155/2015/258619.

[28] Li Y, Zhang H, Shen Q. Spectral–spatial classification of hyperspectral imagery with 3D convolutional neural network. Remote Sens 2017;9(1). http://dx.doi.org/10.3390/rs9010067.

[29] Ienco D, Gaetano R, Dupaquier C, Maurel P. Land cover classification via multitemporal spatial data by deep recurrent neural networks. IEEE Geosci Remote Sens Lett 2017;14(10):1685–9. http://dx.doi.org/10.1109/LGRS.2017.2728698.

[30] Ienco D, Gaetano R, Interdonato R, Ho Tong Minh D. Combining sentinel-1 and sentinel-2 time series via RNN for object-based land cover classification. 2019, p. 4881–4. http://dx.doi.org/10.1109/IGARSS.2019.8898458.

[31] Wu H, Prasad S. Convolutional recurrent neural networks forhyperspectral data classification. Remote Sens 2017;9(3). http://dx.doi.org/10.3390/rs9030298.

[32] Nelsen RB. An introduction to copulas (springer series in statistics). Berlin, Heidelberg: Springer-Verlag; 2006.

[33] Durante F, Sempi C. Principles of copula theory. Chapman and Hall/CRC; 2015.

[34] Joe H, Xu JJ. The estimation method of inference functions for margins for multivariate models. 1996, http://dx.doi.org/10.14288/1.0225985.

[35] Ang A, Chen J. Asymmetric correlations of equity portfolios. J Financ Econ 2002;63(3):443–94. http://dx.doi.org/10.1016/S0304-405X(02)00068-5.

[36] Kilgore R, Thompson D. Estimating joint flow probabilities at stream confluences by using copulas. Transp Res Rec: J Transp Res Board 2011;2262:200–6. http://dx.doi.org/10.3141/2262-20.

[37] Nabaei S, Sharafati A, Yaseen Z, Shahid S. Copula based assessment of meteorological drought characteristics: Regional investigation of Iran. Agricult Forest Meteorol 2019;276–277:1–10. http://dx.doi.org/10.1016/j.agrformet.2019.06.010.

[38] Durante F, Jaworski P. Spatial contagion between financial markets: A copula-based approach. Appl Stoch Models Bus Ind 2010;26:551–64. http://dx.doi.org/10.1002/asmb.799.

[39] Slechan L, Gorecki J. On the accuracy of copula-based Bayesian classifiers: An experimental comparison with neural networks. 2015, p. 485–93. http://dx.doi.org/10.1007/978-3-319-24069-5_46.

[40] Salinas Gutiérrez R, Hernandez-Aguirre A, Rivera M, Villa Diharce E. Using Gaussian copulas in supervised probabilistic classification. 2011, p. 355–72. http://dx.doi.org/10.1007/978-3-642-15534-5_22.

[41] Mercier G, Moser G, Serpico S. Conditional copulas for change detection in heterogeneous remote sensing images. IEEE Trans Geosci Remote Sens 2008;46:1428–41. http://dx.doi.org/10.1109/TGRS.2008.916476.

[42] Rosenblatt M. Remarks on some nonparametric estimates of a density function. Ann Math Stat 1956;27(3):832–7. http://dx.doi.org/10.1214/aoms/1177728190.

[43] Tamborrino C, Mazzia F. On the classification of hyperspectral images with different copula family. In: Proceedings of the 2021 international conference on numerical analysis and applied mathematics. 2021, in press.

[44] Tamborrino C, Interdonato R, Teisserie M. Sentinel-2 Satellite Image time series land cover classification with Bernstein copula approach. Remote Sensing 2022;14(13). http://dx.doi.org/10.3390/rs14133080, https://www.mdpi.com/2072-4292/14/13/3080.

[45] Salinas Gutiérrez R, Hernandez-Aguirre A, Villa Diharce E. Copula selection for graphical models in continuous estimation of distribution algorithms. Comput Statist 2014;29:685–713. http://dx.doi.org/10.1007/s00180-013-0457-y.

[46] Salinas Gutiérrez R, Hernandez-Aguirre A, Villa Diharce E. Dependence trees with copula selection for continuous estimation of distribution algorithms. In: Genetic and evolutionary computation conference, GECCO'11. 2011, p. 585–92. http://dx.doi.org/10.1145/2001576.2001657.

[47] Joe H. Dependence modeling with copulas. first ed.. CRC Press; 2014, p. 480. http://dx.doi.org/10.1201/b17116.

[48] Sathe S. A novel Bayesian classifier using copula functions. 2006, CoRR abs/cs/0611150 arXiv:cs/0611150.

[49] Hofert M, Kojadinovic I, Maechler M, Yan J. Elements of copula modeling with R. Springer Use R! Series; 2018, URL http://www.springer.com/de/book/9783319896342.

[50] Botev Z, Grotowski J, Kroese D. Kernel density estimation via diffusion. Ann Statist 2010;38. http://dx.doi.org/10.1214/10-AOS799.

[51] Alter O, Brown PO, Botstein D. Singular value decomposition for genome-wide expression data processing and modeling. Proc Natl Acad Sci 2000;97(18):10101–6. http://dx.doi.org/10.1073/pnas.97.18.10101.

[52] Wall ME, Rechtsteiner A, Rocha LM. Singular value decomposition and principal component analysis. In: Berrar DP, Dubitzky W, Granzow M, editors. A practical approach to microarray data analysis. Boston, MA: Springer US; 2003, p. 91–109. http://dx.doi.org/10.1007/0-306-47815-3_5.

[53] Brunton SL, Kutz JN. Data-driven science and engineering: Machine learning, dynamical systems, and control. Cambridge University Press; 2019, p. 3–46. http://dx.doi.org/10.1017/9781108380690.

[54] Eckart C, Young G. The approximation of one matrix by another of lower rank. Psychometrika 1936;1:211–8. http://dx.doi.org/10.1007/BF02288367.

[55] Andrew G, Gao J. Scalable training of L^1-regularized log-linear models. In: Proceedings of the 24th international conference on machine learning. ICML '07, New York, NY, USA: Association for Computing Machinery; 2007, p. 33–40. http://dx.doi.org/10.1145/1273496.1273501.

[56] Andrew G, Gao J. Scalable training of L1-regularized log-linear models. In: International conference on machine learning. International Conference on Machine Learning. 2007, URL https://www.microsoft.com/en-us/research/publication/scalable-training-of-l1-regularized-log-linear-models/.

[57] Wang X, Feng Y. New method based on support vector machine in classification for hyperspectral data. In: 2008 International symposium on computational intelligence and design, vol. 1. 2008, p. 76–80. http://dx.doi.org/10.1109/ISCID.2008.61.

[58] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput 1997;9(8):1735–80. http://dx.doi.org/10.1162/neco.1997.9.8.1735.

[59] Margherita G, Enrico B, Giorgio V. Metrics for multi-class classification: an overview. 2020, http://dx.doi.org/10.48550/arXiv.2008.05756, arXiv abs/2008.05756.

[60] McHugh M. Interrater reliability: The kappa statistic. Biochem Med 2012;22:276–82. http://dx.doi.org/10.11613/BM.2012.031.

[61] Roy S, Krishna G, Dubey SR, Chaudhuri B. HybridSN: Exploring 3-D-2-D CNN feature hierarchy for hyperspectral image classification. IEEE Geosci Remote Sens Lett 2019;17:277–81. http://dx.doi.org/10.1109/LGRS.2019.2918719.

[62] Liu S, Luo H, Tu Y, He Z, Li J. Wide contextual residual network with active learning for remote sensing image classification. 2018, p. 7145–8. http://dx.doi.org/10.1109/IGARSS.2018.8517855.

[63] Falini A, Mazzia F, Tamborrino C. Spline based Hermite quasi interpolation for univariate time series. Discrete Contin Dyn Syst - S 2022. http://dx.doi.org/10.3934/dcdss.2022039.