Detection of Sand Dunes on Mars Using a Regular Vine-based Classification Approach

Diana Carrera[1], Lourenço Bandeira[2], Roberto Santana[3], José A. Lozano[4]

Abstract

This paper deals with the problem of detecting sand dunes from remotely sensed images of the surface of Mars. We build on previous approaches that propose methods to extract informative features for the classification of the images. The intricate correlation structure exhibited by these features motivates us to propose the use of probabilistic classifiers based on R-vine distributions to address this problem. R-vines are probabilistic graphical models that combine a set of nested trees with copula functions and are able to model a wide range of pairwise dependencies. We investigate different strategies for building R-vine classifiers and compare them with several state-of-the-art classification algorithms for the identification of Martian dunes. Experimental results show the adequacy of the R-vine-based approach to solve classification problems where the interactions between the variables are of a different nature between classes and play an important role in that the classifier can distinguish the different classes.

Keywords: image dune detection, machine learning, regular vine copula, supervised classification

1. Introduction

Aeolian features are those produced by the action of the wind on a given surface. They are not only found on Earth but have also been reported for other planets and satellites [23, 62]. Analyzing the characteristics of these features can provide information about the current or past atmospheric circulation patterns on the planet; providing therefore relevant information about the atmospheric conditions in the area.

Dunes are the most frequent aeolian features on the Martian surface, and their study contributes to the understanding of the interactions between the atmosphere and the surface of the planet, the way the climate has evolved throughout the history of Mars and how it currently is [32, 63].

In recent years, the need to process large volumes of remotely sensed images has greatly boosted research into the use of automated methods for feature and change detection of structures on planetary surfaces and several works on the analysis of remote sensed imagery have been reported [8, 9, 22, 36, 55].

Recently, supervised learning approaches based on image analysis and pattern recognition techniques have been applied to detect sand dunes on remotely sensed images of the surface of Mars. In particular, support vector machines (SVM) [61] and boosting [65] algorithms were applied to features derived from gradient analysis made at each pixel of the images [3]. However, although a diversified image dataset was used containing examples from both hemispheres of Mars, the dunes present were mainly of the barchan and barchanoid types. In a subsequent work, SVM and random forests [14] were evaluated for this problem with a set of high spatial resolution images including all types of Martian dunes [4].

Inspired by [4], in this work we deal with the dune detection problem, which consists of identifying the presence or absence of sand dunes from images of the surface of Mars whatever their scale. The set of features (variables) –gradient histogram– extracted from these images describes both the directional and periodic characteristics of the dunes, regardless of the diversity of Martian dune types. We use the same feature representation proposed in that

[1]D. Carrera is with the Intelligent Systems Group, Department of Computer Science and Artificial Intelligence, University of the Basque Country, Manuel de Lardizabal, 1, 20018, Donostia, Gipuzkoa, Spain (e-mail: dianamaria.carrera@ehu.eus).

[2]L. Bandeira is with Centre for Natural Resources and the Environment, Instituto Superior Técnico, Av. Rovisco Pais,1049-001 Lisboa, Portugal (e-mail: lpcbandeira@tecnico.ulisboa.pt).

[3]R. Santana is with the Intelligent Systems Group, Department of Computer Science and Artificial Intelligence, University of the Basque Country, Manuel de Lardizabal, 1, 20018, Donostia, Gipuzkoa, Spain (e-mail: roberto.santana@ehu.es).

[4]J. A. Lozano is with the Intelligent Systems Group, Department of Computer Science and Artificial Intelligence, University of the Basque Country, Manuel de Lardizabal, 1, 20018, Donostia, Gipuzkoa, Spain (e-mail: ja.lozano@ehu.es).

work. However, we take another direction using a probabilistic approach to the classification problem based on copula functions.

Among the formalisms used to model dependence structures among random variables, copula functions are one of the most flexible. In simple terms, a copula is a probability distribution function with uniformly distributed margins [46]. Despite the generality of the copula-based framework, building high-dimensional copulas is a difficult problem [2]. Whereas there is a variety of bivariate copula families that cover a wide range of different types of bivariate dependence [38, 46], the class of higher dimensional joint copulas (such as the multivariate Gaussian, Student-t and Archimedean copulas) is quite limited [19].

The pair-copula construction (PCC) method solves this problem by decomposing a multivariate copula in terms of bivariate copulas [5, 6, 37]. These constructions are organized in a graphical way involving a sequence of linked trees called regular vines (R-vines), which makes it easy to identify the pair-copulas associated to the edges of the trees. R-vine copulas are able to model a wide variety of dependencies in multivariate data, including asymmetric and tail dependencies, by combining pair-copulas of different families in the same tree-structure. Other models, such as the multivariate Gaussian copula, assume linear correlation between the variables.

From the machine learning point of view, regular vine copulas are particularly suitable to solve multidimensional classification problems where interactions between the variables play an important role. Previous results in the application of PCCs in classification tasks show that vine copula classifiers can efficiently exploit intricate interactions between the variables at the time of classifying data from different classes [16, 17, 60].

A relevant antecedent of the present work is the proposal made in [16]. The classification approach utilized in that work is based on drawable vine (D-vine) copulas –a particular class of R-vines in which the variables and its associated pair-copulas are organized in a specific and fixed order so that each tree has a path structure [2, 41]. They have a more restrictive structure than R-vines, which can limit their capacity to produce accurate factorizations of distributions.

The classifier introduced in [16] learns a different D-vine copula for each class of the problem. This characteristic also represents a limitation in terms of the interpretability and the potential use of the classifier to reveal insights into the nature of the significant pairwise interactions, and the variability of these interactions among classes. It is common in machine learning classifiers to assume the structure of the models to be the same among classes. This is the case of the naive Bayes classifier (NBC) [50], which assumes attributes are independent of each other, given the class; and more remarkably the tree augmented Bayesian classifier (TAN) [26, 39], which employs a tree structure, in which each attribute only depends on the class and one other attribute. In both classifiers, there is a unique tree structure shared among the classes, although learned with a different method to the one we propose here. Sharing structure sacrifices some accuracy but the task of identifying which pairwise dependencies are those that can help to characterize the classes, and the one of assessing how the strength and shape of the bivariate dependencies changes among the classes, are eased.

In this work, we modify the approach proposed in [16] in two significant ways: (i) We introduce the more general class of R-vines as the models used by the classifiers; (ii) Based on the R-vine learning algorithm developed in [20], we propose two strategies (called CS1 and CS2) that allow us to build a single tree-structure for all classes while pair-copulas are selected and estimated from the data of the corresponding class. Although this constraint leads to a lack of flexibility of the model, and therefore deteriorates the performance of the classifiers, the experimental results show that the impact is not too severe due to the fact that R-vine classifiers with shared structure still keep a high degree of flexibility thanks of using pair-copulas of different families that are estimated from the data of the corresponding class. This strategy also contributes to reduce the cost of construction of R-vine classifiers because only a single tree-structure is built for all classes instead of one for each class.

We also show the feasibility of R-vine copulas to deal with a real-world high dimensional problem. The first papers reporting R-vine-based applications addressed problems with a small number of variables, for instance [2, 58], however, this situation has changed. Recent research reports R-vines results on larger problems. For instance, 448 variables in [15], 408 in [16], 400 in [44], and 96 in [45]. The classification problem we address in this paper has 180 variables (attributes or features).

The remainder of this paper is organized as follows: Section 2 presents the general PCC method for multivariate copulas using only pair-copulas, including the derivation of R-vine models. Section 3 describes the procedure for learning R-vine distributions that constitute R-vine classifiers. Section 4 discusses the general vine copula classification approach utilized in this work to construct R-vine classifiers. Two learning methods that guarantee the same tree-structure for all classes of an R-vine classifier are introduced in Section 5. The automatic dune detection problem, along with a brief description of the methods applied for feature extraction, are given in Section 6. Section 7 presents the experimental framework and discusses the numerical results. Section 8 concludes with a summary and an outlook to possible future research.

2. R-vine Copulas

In this section we offer both a brief introduction to copulas and a review of the pair-copula construction (PCC) method and R-vine copula models following [2, 19].

2.1. Copulas

Let $\mathbf{X} = (X_1, \ldots, X_n)$ be an $n-$dimensional random vector with joint density function $f : \mathbb{R}^n \to [0, \infty)$ and cumulative distribution function $F : \mathbb{R}^n \to [0, 1]$. Furthermore, let $F_i : \mathbb{R} \to [0, 1]$, $i = 1, \ldots, n$ be the corresponding marginal distributions of X_i[5]. Capital letters denote variables, whereas lower case letters are their assignments.

A copula is a multivariate probability distribution function for which the marginal probability distribution of each variable is uniform [46]. Copulas are used to describe the dependence structure among random variables.

The formal definition of a copula is as follows [46]: $C : [0, 1]^n \to [0, 1]$ *is an n-dimensional copula if* C *is a joint cumulative distribution function of an n-dimensional random vector on the unit cube* $[0, 1]^n$ *with uniform marginals.*

The importance of copulas in probabilistic modeling is given by the Sklar's theorem [57], which states that an n-dimensional (multivariate) cumulative distribution function $F(x_1, \ldots, x_n)$ of a random continuous vector $\mathbf{X} = (X_1, \ldots, X_n)$ can be expressed in terms of its marginal distributions $F_i(x_i)$ and a unique copula C as follows

$$F(x_1, \ldots, x_n) = C(F_1(x_1), \ldots, F_n(x_n)) \tag{1}$$

with $(x_1, \ldots, x_n) \in \mathbb{R}^n$. For an absolutely continuous F with strictly increasing, continuous marginals F_i, the corresponding joint density function f is given by

$$f(x_1, \ldots, x_n) = c(F_1(x_1), \ldots, F_n(x_n)) \cdot \prod_{i=1}^{n} f_i(x_i) \tag{2}$$

where c and f_i denote the corresponding density functions of C and F_i respectively. Copulas allow to model and estimate the distribution of random vectors by means of the separate estimation of its marginals and the copula that represents the dependence structure between the variables.

The most simple copula is the product copula. It appears naturally as the copula in (1) associated to a random vector of independent variables. It can be shown that if $F(x_1, \ldots, x_n) = F_1(x_1) \cdot \ldots \cdot F(x_n) = C(F_1(x_1), \ldots, F_n(x_n))$, then C is defined as $C(u_1, \ldots, u_n) = u_1 \cdot \ldots \cdot u_n$ and $c(u_1, \ldots, u_n) = 1$.

The multivariate Gaussian copula is the copula associated to the multivariate standard normal distribution. In the bivariate case, for example, departing from Sklar's theorem (2) we get

$$f(x_1, x_2) = c_{12}(F_1(x_1), F_2(x_2)) \cdot f_1(x_1) \cdot f_2(x_2), \tag{3}$$

with

$$f_i(x_i) = \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{1}{2}x_i^2\right\}, \; i = 1, 2 \tag{4}$$

and

$$f(x_1, x_2) = \frac{1}{2\pi\sqrt{(1-\rho_{12}^2)}} \cdot \exp\left\{-\frac{1}{2(1-\rho_{12}^2)}\left(x_1^2 + x_2^2 - 2\rho_{12}x_1x_2\right)\right\} \tag{5}$$

the univariate and bivariate densities, respectively, where ρ_{12} is the correlation between X_1 and X_2.

Using (3) we have [56]

$$\frac{f(x_1, x_2)}{f_1(x_1) f_2(x_2)} = c_{12}(F_1(x_1), F_2(x_2)) \tag{6}$$

and therefore

$$c_{12}(u_1, u_2) = \frac{1}{1-\rho_{12}^2} \cdot \exp\left\{-\frac{1}{2(1-\rho_{12}^2)}\left[\rho_{12}^2\left(x_1^2 + x_2^2\right) - 2\rho_{12}x_1x_2\right]\right\} \tag{7}$$

where $u_1 = \phi_1(x_1)$, $u_2 = \phi_2(x_2)$, and $x_1 = \phi_1^{-1}(u_1), x_2 = \phi_2^{-1}(u_2)$ are the inverse of the standard univariate Gaussian distribution function.

[5] We assume that all multivariate, marginal and conditional distributions are absolutely continuous with corresponding densities.

2.2. Pair-copula Constructions

In many real-world situations the type of relations between n random variables can not be modeled with single standard multivariate copulas (e.g., the Gaussian, Student-t or Archimedean copulas). These copulas express the same type of dependence between all sets of variables, which is too restrictive when the variables are correlated in different ways. Pair-copula constructions (PCCs) [5, 6, 37] solve this matter by constructing a model which uses bivariate copulas as building blocks.

The modeling scheme of PCCs is based on a decomposition of a multivariate density into a cascade of (conditional and unconditional) bivariate copulas called pair-copulas [2]. PCCs exploit the fact that the bivariate copulas are more tractable than multidimensional copulas. They provide a very flexible way of dependence modeling since pair-copulas belonging to different families can be combined in the same decomposition [19], accommodating in this way complex dependence structures such as weak/strong tail behavior and symmetric/asymmetric dependence. A comprehensive reference about bivariate copula families can be found in [38, 46].

The starting point of PCC method for constructing multivariate distributions is the chain rule

$$f(x_1, \ldots, x_n) = f(x_1) \cdot f(x_2 \mid x_1) \cdot f(x_3 \mid x_1, x_2) \cdot \ldots \cdot f(x_n \mid x_1, \ldots, x_{n-1}) \tag{8}$$

where each conditional density can be decomposed into a pair-copula and a conditional marginal density as follows:

$$f(x_i \mid \mathbf{x_v}) = c_{ij|\mathbf{v}_{-j}} \left(F\left(x_i \mid \mathbf{x}_{\mathbf{v}_{-j}}\right), F\left(x_j \mid \mathbf{x}_{\mathbf{v}_{-j}}\right)\right) \cdot f\left(x_i \mid \mathbf{x}_{\mathbf{v}_{-j}}\right) \tag{9}$$

for a d-dimensional vector $\mathbf{x_v}$, where x_j is one arbitrarily chosen component of $\mathbf{x_v}$, and $\mathbf{x}_{\mathbf{v}_{-j}} = \mathbf{x_v} \setminus x_j$ denotes the vector $\mathbf{x_v}$ excluding the component x_j.

For instance, the second factor of (8) is the simplest conditional term and can be written using (9) as follows:

$$f(x_2 \mid x_1) = c_{12}\left(F_1(x_1), F_2(x_2)\right) f_2(x_2). \tag{10}$$

In the three-variate case, the third factor of (8) can be decomposed for the pair-copula $c_{13|2}$ and treated as a bivariate density again as

$$f(x_3 \mid x_1, x_2) = c_{13|2}\left(F(x_1 \mid x_2), F(x_3 \mid x_2)\right) f(x_3 \mid x_2). \tag{11}$$

Notice that we could also have decomposed the third term for the pair-copula $c_{23|1}$ as

$$f(x_3 \mid x_1, x_2) = c_{23|1}\left(F(x_2 \mid x_1), F(x_3 \mid x_1)\right) f(x_3 \mid x_1) \tag{12}$$

where $c_{13|2}$ is different from $c_{23|1}$. So, given a specific factorization, there are different decompositions. Decomposing $f(x_3 \mid x_2)$ in (11) further leads to

$$f(x_3 \mid x_1, x_2) = c_{13|2}\left(F(x_2 \mid x_1), F(x_3 \mid x_1)\right) \cdot c_{32}\left(F(x_3), F(x_2)\right) \cdot f(x_3) \tag{13}$$

where two pair-copulas, c_{32} and $c_{13|2}$, are presented.

In (9), the conditional distributions $F(x_i \mid \mathbf{v}_{-j})$ and $F(x_j \mid \mathbf{v}_{-j})$ that constitute the arguments of pair-copulas can be obtained for every x_j in $\mathbf{v}$ by recursively applying the relationship [37],

$$F(x_i \mid \mathbf{x_v}) = \frac{\partial C_{ij|\mathbf{v}_{-j}}\left(F\left(x_i \mid \mathbf{x}_{\mathbf{v}_{-j}}\right), F\left(x_j \mid \mathbf{x}_{\mathbf{v}_{-j}}\right)\right)}{\partial F\left(x_j \mid \mathbf{x}_{\mathbf{v}_{-j}}\right)}. \tag{14}$$

Note that a general probability distribution does not need to be expressed as a PCC because the copula $C_{ij|\mathbf{v}_{-j}}$ (9) is the same for each value of the conditioning set $\mathbf{x}_{\mathbf{v}_{-j}}$, and, of course, different values of $\mathbf{x}_{\mathbf{v}_{-j}}$ can produce different joint distributions between $x_i, x_j \mid \mathbf{x}_{\mathbf{v}_{-j}}$. In some cases, however, a probability distribution can be factorized as a PCC through the so called simplified assumption [59]. The simplifying assumption assumes that the copula describing the dependence of two variables given a set of variables does not depend on the specific values of the conditional values, however the R-vine density depends on the conditioning values through the arguments of the bivariate copula. Moreover, even when the PCC factorization is not exact, it can be a good approximation of the probability distribution.

2.3. R-vine Distributions

In high dimensions, there exist many PCC decompositions. To organize such decompositions, a graphical model called regular vine (R-vine) was introduced in [5, 6]. It involves the specification of a sequence of trees, each edge of which corresponds to a pair-copula. These pair-copulas constitute the building blocks of the multivariate R-vine copula.

An R-vine is a probabilistic graphical model that is represented as a pair (T, θ). T is the structural part that is composed of a sequence of trees $T_1, T_2, \ldots, T_{n-1}$, where the nodes of T_j are the edges of T_{j-1}. θ contains, for each edge of the trees, a pair-copula and its associated parameters [19, 41]. This design allows the R-vine tree-structure to have pair-copulas of different families.

To specify an n-dimensional distribution with an R-vine, $n-1$ trees and $\frac{n(n-1)}{2}$ pair-copulas are necessary. Fig. 1 shows the tree-structure of a 5-dimensional R-vine. We can see that the first tree T_1 requires four unconditional copulas: c_{12}, c_{23}, c_{34}, c_{35}; T_2, three conditional copulas: $c_{13|2}$, $c_{24|3}$, $c_{25|3}$; T_3, two conditional copulas: $c_{15|23}$, $c_{14|23}$; and the last tree T_4, one conditional copula: $c_{45|123}$. In general, in T_1, $n-1$ unconditional copulas are needed, whereas in the subsequent levels $j = 2, \ldots, n-1$, each tree T_j requires $n-j+1$ conditional copulas, one less in each level.

If we denote $T_j = (N_j, E_j)$, the tree of the decomposition in the level j, where N_j and E_j denote the node and edge sets of the j^{th} tree, the edge $e \in E_j$ joins two vertices of N_j, which are represented by $k(e)$ and $l(e)$. Then, in the corresponding pair-copula, the nodes $k(e)$ and $l(e)$ are the conditioned nodes and $D(e)$ is the conditioning set. Let $\mathbf{X}_{D(e)}$ be the subvector of $\mathbf{X}$ determined by the indices contained in $D(e)$. Then, a *regular vine distribution* is the distribution of the random vector $\mathbf{X} = (X_1, \ldots, X_n)$ with marginal densities $f_i(x_i)$, $i = 1, \ldots, n$, and the conditional density of $\left(X_{k(e)}, X_{l(e)}\right)$ given the variables $\mathbf{X}_{D(e)}$ specified as $c_{k(e),l(e)|D(e)}$ for the R-vine copula with $n-1$ trees with set of nodes $\mathbf{N} = \{N_1, \ldots N_{n-1}\}$ and set of edges $\mathbf{E} = \{E_1, \ldots E_{n-1}\}$ [19].

As shown in [19, 42], if the random vector $\mathbf{X}$ follows an R-vine distribution, there is a unique density $f(x_1, \ldots, x_n)$ of X as follows:

$$\prod_{j=1}^{n-1} \prod_{e \in E_j} c_{k(e),l(e)|D(e)} \left(F\left(x_{k(e)} \mid \mathbf{x}_{D(e)}\right), F\left(x_{l(e)} \mid \mathbf{x}_{D(e)}\right)\right) \cdot \prod_{i=1}^{n} f_i(x_i) \tag{15}$$

where $\mathbf{x}_{D(e)}$ denotes the subvector of $\mathbf{x}$ determined by indices in $D(e)$ [13, 41].

3. Learning of R-vine Distributions

An attractive feature of the copulas is that they provide a flexible tool to easily model and estimate the distribution of random vectors by estimating marginals and copulas separately. This feature is used to design learning procedures where the estimation of the margins and the R-vine copula is usually conducted in two steps: the first step estimates the margins and their parameters and the second step estimates the R-vine copula. An algorithm for learning R-vine copulas is developed in [20]. This algorithm must complete the following three main tasks: (i) selection of the R-vine hierarchical tree structure, (ii) selection of pair-copula family, and (iii) estimation of their parameters.

Steps listed in Algorithm 1 allow to build a whole R-vine distribution of an n-dimensional vector $(X_1, \ldots, X_n)$: margins + the R-vine copula. In this algorithm, Steps 3-7 correspond to the learning procedure of R-vine copulas proposed in [20]. Step 6 is optional and corresponds to a strategy developed in [13] for simplifying R-vines that yields the so called truncated R-vine. Before explaining Algorithm 1, we introduce the necessary notation. The input of the algorithm is an $S \times n$ matrix (original data) $\mathbf{D}_{\mathbf{X}} = (\mathbf{x}_1, \ldots, \mathbf{x}_S)^T$ consisting of S (sample size) n-dimensional vectors $\mathbf{x}_1 = (x_{11}, \ldots, x_{1n}), \ldots, \mathbf{x}_S = (x_{S1}, \ldots, x_{Sn})$ which are observations (instances) of the random vector $\mathbf{X} = (X_1, \ldots, X_n)$. Besides, the uniform (unconditional or conditional) copula data required at level j of the R-vine decomposition is an $S \times (n-j+1)$ matrix $\mathbf{D}_{\mathbf{U}} = (\mathbf{u}_1, \ldots, \mathbf{u}_S)^T$ consisting of S $(n-j+1)$-dimensional vectors $\mathbf{u}_1 = \left(u_{11}, \ldots, u_{1(n-j+1)}\right), \ldots, \mathbf{u}_S = \left(u_{S1}, \ldots, u_{S(n-j+1)}\right)$ which are observations of the random vector $\mathbf{U} = (U_1, \ldots, U_{n-j+1})$. Indices $s = 1, \ldots, S$, $i = 1, \ldots, n$ and $j = 1, \ldots, n-1$ run through the samples, variables and trees respectively.

Algorithm 1 starts by estimating n marginal cumulative distributions $F_i(x_i)$ from the original data $\mathbf{D}_{\mathbf{X}}$. Then, by evaluating $u_i = F_i(x_i)$ it compute the unconditional copula needed to learn the R-vine copula of $f(x_1, \ldots, x_n)$. To estimate the R-vine copula, the stepwise tree-by-tree procedure proposed in [20] is applied : To select the first tree, the maximum weighted spanning tree (MWST) is computed, where the absolute values of Kendall's tau are used as

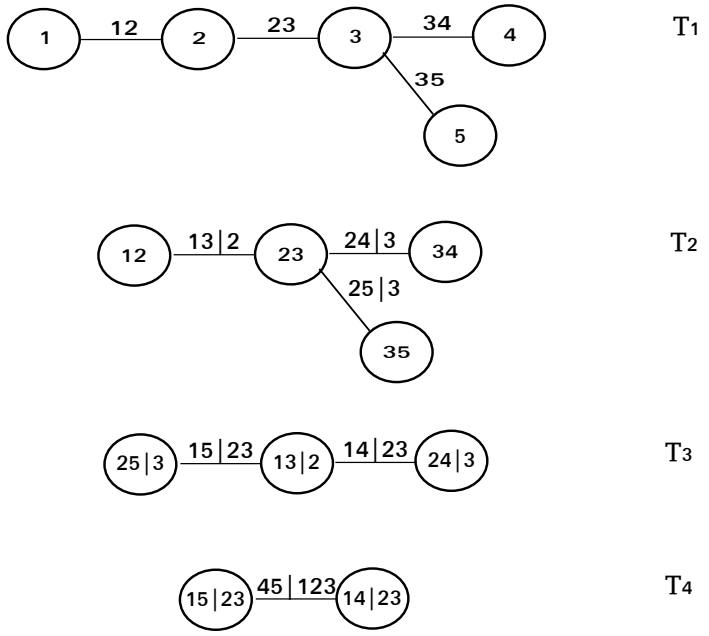


Figure 1: Example of a 5-dimensional R-vine structure. The corresponding R-vine distribution has the joint density $f(x_1, \ldots, x_5)$ given by $\underbrace{c_{12} \cdot c_{23} \cdot c_{34} \cdot c_{35}}_{T_1} \cdot \underbrace{c_{13|2} \cdot c_{24|3} \cdot c_{25|3}}_{T_2} \cdot \underbrace{c_{15|23} \cdot c_{14|23}}_{T_3} \cdot \underbrace{c_{45|123}}_{T_4} \cdot \prod_{i=1}^{5} f_i(x_i)$.

weights of the edges; i.e., it maximizes the sum of pairwise dependencies. Given this tree, the procedure selects pair-copulas, estimates their parameters and computes transformed observations $F\left(x_{k(e)} \mid \mathbf{x}_{D(e)}\right)$ and $F\left(x_{l(e)} \mid \mathbf{x}_{D(e)}\right)$ (i.e., the arguments of the pair-copulas), which constitute the unconditional copula data needed to estimate the next tree. The procedure executed for the first tree (or level) is repeated tree-by-tree with the additional restriction that the set of possible edges in T_j, $j = 2, \ldots, n-1$, must satisfy the *proximity condition* [19], which states that the linking nodes in tree T_j share a common node in the tree above T_{j-1}. Steps 3-7 are repeated until all trees are built and their pair-copulas estimated. Notice that the conditioning set $D(e)$ associated to each pair-copula $c_{k(e),l(e)|D(e)}$ increases by one variable at each new level.

Next, we offer a detailed description of the steps of Algorithm 1 according to the particular implementation used in this work[6]:

- Step 1 The univariate cumulative and density functions $F_i(x_i)$ and $f(x_i)$ are estimated from the original data matrix $\mathbf{D_X}$. Two strategies can be used to select the type of margins: The simplest, but rather inaccurate, strategy selects the same family of distributions to model all margins. A more flexible method chooses, among a predefined group of families, the one closest to the empirical univariate distribution.

- Step 2 Using the distributions learned in Step 1, we calculate the (unconditional) copula data observations $\mathbf{u}_s = (u_{s1}, \ldots, u_{sn})$, $s = 1, \ldots, S$, where $u_{si} = F_i^{-1}(x_{si})$.

- Step 3 The algorithm uses the Kendall's tau coefficient $-1 \leq \tau \leq 1$ to measure the strength of (monotonic) dependence between two copula variables U and V (Kendall's tau does not rely on any assumptions on the distributions of the variables). The higher the absolute Kendall's tau value is, the higher the association between U and V. If the variables are independent, then we expect the coefficient to be approximately zero. Given the pairs (u_i, v_i) and (u_j, v_j), they are concordant if $(u_i - u_j)(v_i - v_j) > 0$ and discordant if $(u_i - u_j)(v_i - v_j) < 0$. The case $(u_i - u_j)(v_i - v_j) = 0$ occurs with probability zero under the assumption that the variables are continuous [27]. It is calculated from data as $\frac{n_c - n_d}{n(n-1)/2}$ where n_c and n_d denote the total

[6]The implementation of the algorithm is available from https://github.com/DianaCarrera

Algorithm 1 Learning procedure of an R-vine distribution.

INPUT: $\mathbf{D_X} = (\mathbf{x}_1, \ldots, \mathbf{x}_S)^T$ – Original data.
OUPUT: An R-vine distribution.

Step 1. Estimate the univariate cumulative and density functions $F_i(x_i)$ and $f_i(x_i)$ from $\mathbf{D_X}$.
Step 2. Compute $S \times n$ observations $u_{si} = F_i(x_{si})$ to obtain the copula data $\mathbf{D_U}$.
for $j = 1 : n-1$
  Step 3. Calculate the Kendall's tau for all pairs that are allowed by the *proximity condition*.
  Step 4. Select the MWST T_j that satisfies the *proximity condition* over $n - j + 1$ nodes.
  Step 5. Select pair-copulas and estimate their parameters.
  Step 6. If $j \neq 1$, perform the BIC-based truncation strategy to choose the truncation level.
    if $BIC_j \geq BIC_{j-1}$ **then**
      Truncated at level $j-1$. In (15), all pair-copulas of order$\geq j$ are set as product copulas.
      break for
    end if
  Step 7. Compute $S \times (n - j + 1)$ transformed observations using (14) to obtain the new $\mathbf{D_U}$.
end for

number of concordant and discordant pairs, respectively, n is the sample size, and $n(n-1)/2$ is the total number of possible pairings of U and V observations. Notice that only those pairs allowed by the *proximity condition* are considered.

- Step 4 The tree-structure is chosen via a MWST algorithm that maximizes the sum of the absolute empirical Kendall's taus of all possible pairs (those that satisfy the *proximity condition*) on a given level. In the first level $j = 1$, the MWST algorithm acts on n variables, while in the subsequent levels $j = 2, \ldots, n-1$, it acts on $n - j + 1$ variables (one less in each level of depth).

- Step 5 In general, one can select the same copula family for all edges and, in this case, we only need to estimate their parameters as explained later. More flexible approaches for copula selection and estimation based on maximum likelihood (MLE) have been proposed in the literature [7, 12, 28, 52]. The approach used in this work consists in selecting the suitable pair copula family individually for each edge $e \epsilon E_j$ of T_j by comparing a parametric copula C_θ (chosen from a predefined subgroup of families) to the empirical copula $C_\mathbf{e}$ [27], which is given in [1] by

$$C_\mathrm{e}(u, v) = \frac{1}{S} \sum_{s=1}^{S} \mathbf{1}(u_s \leq u, v_s \leq v) \tag{16}$$

where S is the sample size, $0 < u, v < 1$. The tested copula C_θ (with parameter θ) closest to the empirical distribution C_e of the bivariate data is chosen using a measure of distance given by

$$d(C_\mathrm{e}, C_\theta) = \sqrt{\sum_{s=1}^{S} \left(C_\mathrm{e}(u_s, v_s) - C_\theta(u_s, v_s)\right)^2}. \tag{17}$$

Having selected the appropriate copula families using 17, the algorithm proceeds to estimate the corresponding copula parameters via the Kendall's tau-based method when the copula parameter is a scalar [40]. The relationship of Kendall's tau and the dependence parameter of the bivariate Gaussian, Student's t, Clayton and Gumbel bivariate families are given in Table 1. Of these copulas, only the Student's t has two parameters: ρ and the degrees of freedom ν, which capture the amount of tail dependence.

- Step 6 The computational effort required to estimate R-vine copulas increases with the dimension. To counteract this problem, the algorithm applies the truncation strategy developed in [13], which is based on the Bayesian Information Criterion (BIC) [54]. By replacing certain pair-copulas by the independence copula, this strategy allows to simplify the tree-structure selecting the truncation level by checking if the following condition is met: If the metric calculated up to the tree at level j is greater than or equal to the metric up

Table 1: Relationship of Kendall's tau and the parameter of different families of bivariate copulas.

Bivariate copula	Parameter	Kendall's tau
Gaussian	$\rho = \sin\left(\frac{\pi}{2}\tau\right),\ \rho \in (-1,1)$	$\frac{2}{\pi} arcsin\left(\rho\right)$
Student's t	$\rho = \sin\left(\frac{\pi}{2}\tau\right),\ \rho \in (-1,1)$	$\frac{2}{\pi} arcsin\left(\rho\right)$
Clayton	$\delta = 2\tau/\left(1-\tau\right),\ \delta > 0$	$\frac{\delta}{\delta+2}$
Gumbel	$\delta = 1/\left(1-\tau\right),\ \delta \geq 1$	$1 - \frac{1}{\delta}$

to the previous level $j-1$, we keep the tree T_j; otherwise, the R-vine tree-structure is truncated at T_{j-1} and all the pair-copulas in the subsequent trees are assumed to be product copulas.

- Step 7 The last step of Algorithm 1 consists of obtaining the (conditional) copula data needed to select the next tree. The transformed observations are obtained by the recursive evaluation of conditional distributions functions $F\left(x_{k(e)} \mid \mathbf{x}_{D(e)}\right)$ and $F\left(x_{l(e)} \mid \mathbf{x}_{D(e)}\right)$ defined in (14).

Appendix provides an "execution" of Algorithm 1 step by step through a simple example of four variables.

4. Building R-vine Classifiers

When using probabilistic models for supervised classification, a natural approach is to learn a model for each class of the problem. This was the approach proposed in [16] where D-vines were introduced in the context of supervised classification. We follow a similar approach: an R-vine classifier learns one R-vine distribution $p^k\left(\mathbf{x}\right)$ for each class label $k = 1, \ldots, K$ from the corresponding training instances (observations). Then, the learned models are used to predict the most likely class label $\hat{k}$ of the unlabeled sample $\mathbf{x}$ by applying a decision rule, which is formulated as $\hat{k} = \max_{k \in \{1,\ldots,K\}} p^k\left(\mathbf{x}\right) \cdot p\left(k\right)$, where $p\left(k\right)$ is the probability of the class k.

With regard to the components that determine the complexity of the classifier - namely, the number of trees, the families of pair-copulas, and the types of margins, we define three groups of classifiers (the present taxonomy extends that previously introduced in [16]):

- *Unmixed classifiers:* All R-vines have the same number of trees. They also have all pair-copulas of the same family of copulas, and all margins of the same type of univariate distribution.
- *Partially-mixed classifiers:* Allow for more flexible classifiers than those listed above as far as pair-copulas of different families can be mixed in the same R-vine copula (details in Section 3-Step 5).
- *Fully-mixed classifiers:* They are the most flexible classifiers of this group mixing pair copulas and margins of different families.

5. Learning R-vine Classifiers with a Common Tree-structure

Algorithm 1 describes the modeling scheme used to learn the R-vine distributions used by the R-vine classifier. In this scheme, for each class, an R-vine distribution is learned from instances (data) aiming to get the best fit to the data. However, higher accuracy has a cost, the price that must be paid to build more accurate models is an increase of the cost of the estimation procedure. Learning an R-vine distribution for each class not only may impact the cost of estimation, but, since each class has its own model, also makes it more difficult to identify the most informative features and relevant pairwise interactions of the problem. This is particularly true for problems with a high number of variables. In order to alleviate these disadvantages, we propose two different methods that allow to learn a common R-vine tree-structure for all classes. This constraint limits the flexibility of the learned distributions and therefore impacts the performance of the classifiers. However, this influence could be mitigated because the copulas corresponding to the vines learned in each class are selected using the respective data. Therefore, the difference in the distributions between classes is captured by these copulas.

Let us describe the proposed methods –namely, common structure CS1 and CS2 respectively. These methods learn a common tree-structure that is shared by all classes instead of learning a structure for each class. From now on, this last case is denoted as DS method (acronym for Different Structures). Whereas the DS method executes Algorithm 1 for each class independently, CS1 and CS2 implement a modification of this algorithm since some steps require information coming from all classes.

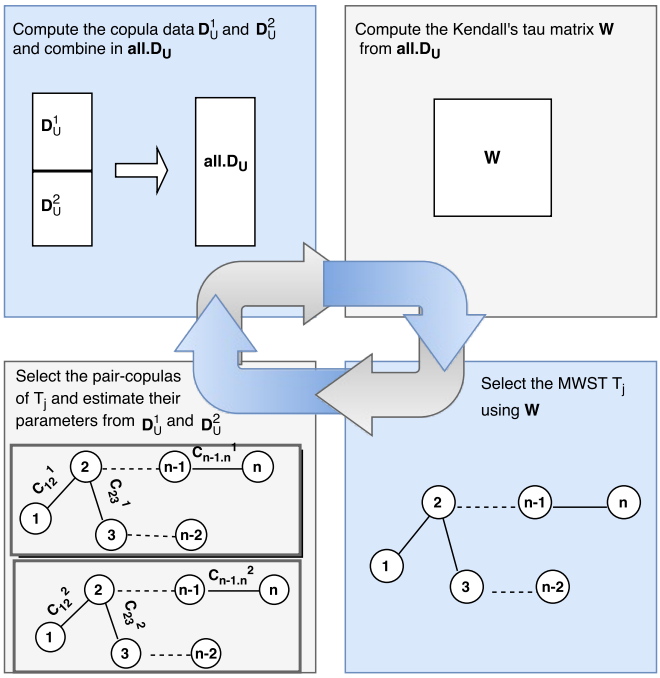


Figure 2: Diagram illustrating how CS1 works. It learns a single tree-structure that is shared by all the classes.

Let us introduce the necessary notation. Let $k = 1, \ldots, K$ stand for the index of K classes, $\mathbf{D}_{\mathbf{X}}^k$ is an $S^k \times n$ matrix that stores the original data belonging to the class k such that $\mathbf{D}_{\mathbf{X}}^k = \left(\mathbf{x}_1^k, \ldots, \mathbf{x}_{S^k}^k\right)^T$, $s = 1, \ldots, S^k$ (S^k denotes the sample size of the original data belonging to the class k) and where $\mathbf{x}_1^k = \left(x_{11}^k, \ldots, x_{1n}^k\right), \ldots, \mathbf{x}_{S^k}^k = \left(x_{S^k1}^k, \ldots, x_{S^kn}^k\right)$ denote observations of the n-dimensional random vector $\mathbf{X} = (X_1, \ldots, X_n)$. The same rules apply to denote the (unconditional and conditional) copula data for the class k. The copula data for the class k is an $S^k \times (n-j+1)$ matrix $\mathbf{D}_{\mathbf{U}}^k = \left(\mathbf{u}_1^k, \ldots, \mathbf{u}_{S^k}^k\right)^T$ conformed by S^k $(n-j+1)$-dimensional vectors $\mathbf{u}_1^k = \left(u_{11}^k, \ldots, u_{1(n-j+1)}^k\right), \ldots, \mathbf{u}_{S^k}^k = \left(u_{S^k1}^k, \ldots, u_{S^k(n-j+1)}^k\right)$ which are instances (observations) of the random vector $\mathbf{U}^k = \left(U_1^k, \ldots, U_{n-j+1}^k\right)$. Indices $s = 1, \ldots, S^k$, $i = 1, \ldots, n$ and $j = 1, \ldots, n-1$ run over the samples, variables and trees respectively.

5.1. Method CS1

This method consists of learning K R-vine distributions (using the Algorithm 1) with a common tree-structure, where each tree T_j is learned from a single dataset (denoted as $\mathbf{all.D_U}$). This dataset contains K copula data obtained when the R-vine distribution for the class k is learned (at Step 2 if $j = 1$, and at Step 6 for the subsequent trees). $\mathbf{all.D_U}$ is created by combining (by rows) the copula data $\mathbf{D}_{\mathbf{U}}^k$ such that $\mathbf{all.D_U} = \left(\mathbf{D}_{\mathbf{U}}^{\mathbf{1}}, \ldots, \mathbf{D}_{\mathbf{U}}^K\right)_{\left(\sum_{k=1}^K S^k\right) \times n}$. Then, Kendall's taus are computed ($\mathbf{W}$ matrix) from $\mathbf{all.D_U}$ for all possible pairs of variables in the j^{th} level (one variable less at each new level). Next, a single MWST that maximizes the absolute values of previously computed Kendall's taus is found (Step 4). For each class, pair-copulas are selected individually from the corresponding copula data $\mathbf{D}_{\mathbf{U}}^k$ (Step 5). CS1 is executed tree-by-tree until all trees are built and their pair-copulas estimated. The final R-vine tree-structure is shared by all classes. The output of CS1 is K R-vine distributions that share a common tree structure. Fig. 2 shows a diagram describing the steps of method CS1.

When CS1 works together the BIC-based truncation method (see Algorithm 1-Step 6), it may happen that the truncation level is reached at a different level in each class. The question that arises is how to create a common tree-structure for all classes. We propose the following strategy: Suppose we have two classes, namely c1 and c2, and that for each class the truncation level is reached in the trees T_2 and T_5 respectively. For this example, the single tree-structure created with CS2 has five trees. Then, in the last two trees of c1 (i.e., T_4 and T_5) only product copulas are fitted, whereas in the other class, pair-copulas are selected normally from the corresponding copula data.

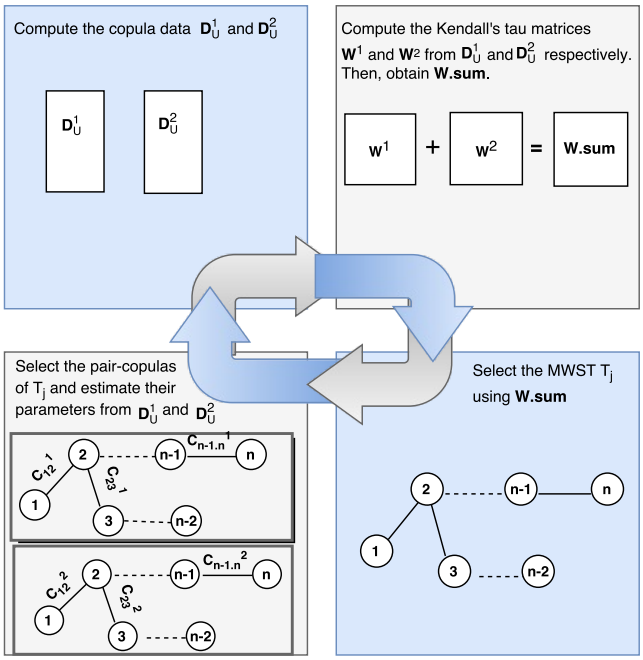


Figure 3: Diagram illustrating how CS2 works. It learns a single tree-structure that is shared by all the classes.

5.2. Method CS2

As CS1, the method CS2 learns K R-vine distributions (using the Algorithm 1) with a common tree-structure, but in a different way. CS2 consists of learning a MWST from a single pairwise weight matrix, denoted as $\mathbf{W}.\mathbf{sum}$. This matrix is computed as follows: Firstly, for each class k, a matrix of absolute Kendall's taus, denoted as $\mathbf{W}^k$, is computed in each level of the R-vine tree-structure from the corresponding copula data $\mathbf{D}_\mathbf{U}^k$. Then, the matrix $\mathbf{w}.\mathbf{sum}$ is obtained by element-wise addition of all $\mathbf{W}^k$ matrices such that $\mathbf{W}.\mathbf{sum} = \sum_{k=1}^{K} \mathbf{W}^k$. Then, for the current level, a single MWST is built using $\mathbf{W}.\mathbf{sum}$ as the weight matrix (Step 4). For each class, pair-copulas are selected individually from the corresponding copula data $\mathbf{D}_\mathbf{U}^k$ (Step 5). CS2 is executed tree-by-tree and until all trees are built and their pair-copulas estimated. The final R-vine tree-structure is shared by all classes. The output of CS2 is K R-vine distributions that share a common tree structure. Fig. 3 shows a diagram describing the steps of method CS2. Similar to CS1, CS2 can also work together the BIC-based truncation strategy to create a common tree-structure.

6. Dune Classification Problem

A geological classification scheme of sand dunes was proposed in [43] for terrestrial examples, mostly based on field work. So far, the dunes identified on the Martian surface have been classified according to that scheme, and although most of them fit into the main types there are some undefined morphologies not known to occur on Earth [34].

Examples of the great diversity of types of Martian dunes can be seen in Fig. 4, barchan, barchanoid, transverse, dome, linear, and star, being among the most representative [34]. From this, it becomes clear the multitude of factors that affect the visual aspect of dune fields –constituents, size, shape and density, association to seasonal advance and withdrawal of ice cover, and angle of illumination, among others– and the need to design classification strategies capable of detecting the presence of dunes on images. This research inspired by [4] represents a further step in this direction.

The dune classification (or detection) problem (DCP) we deal with consists of identifying the presence or absence of sand dunes from remotely sensed images of the surface of Mars. In order to solve this problem, support vector machines [61] and random forests (RF) [14] were used with good results in [4]. Here, we propose a different approach, testing probabilistic classifiers based on R-vine copulas using the same methodology and feature representation introduced in that work. The following sections delve into these aspects.

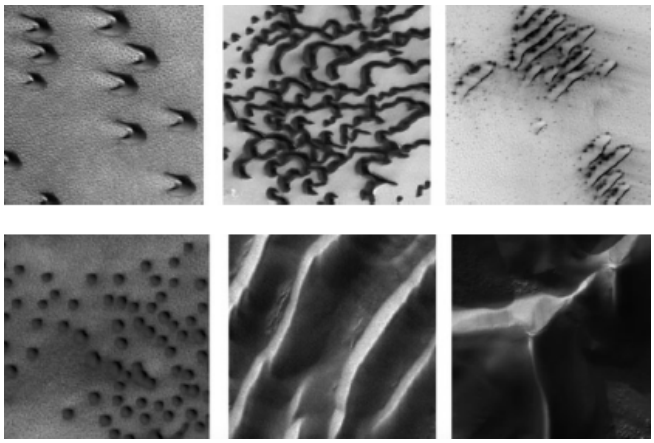


Figure 4: Examples of the diversity of Martian sand dunes (the side of each square image is $2500m$): From left to right and top to bottom: barchan, barchanoid, transverse, dome, linear, and star. Image credits: NASA/JPL/MSSS.

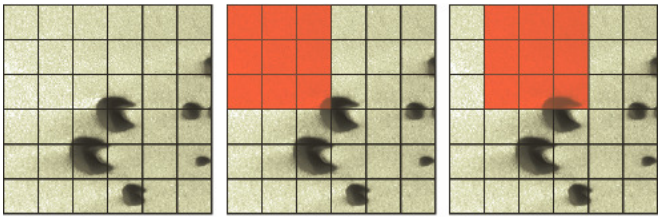


Figure 5: Example of the image analysis: Tiling an image in (left) cells of 40×40 pixels and (center) blocks of 3×3 cells; (right) block displacement with overlapping to the right.

6.1. Methodology for DCP

The methodology adopted for the automatic detection of sand dunes on images is based on the classification of small square regions of the image called cells. The task of extracting and analyzing local information (image features) is done throughout a regular grid of tiled image. The image is divided into cells with 40×40 pixels from which features are extracted (see Fig. 5 - (left)). The way of computing the vectors of features is explained later in Section 6.2.

The size of each cell is the same for all images. To increase the invariance to specific factors such as illumination and shadowing, an aggregation of the local features is performed within larger regions of 3×3 cells called blocks (i.e., one block has nine cells), which are the detection windows (see Fig. 5 - (center)). To analyze the complete image, this block window is moved along the entire image grid with an overlapping between adjacent blocks equal to one cell side (see Fig. 5 - (right)).

Each cell –represented by a vector of features extracted in the block that contains this cell in the center– is classified as dune or non-dune. The labelling is carried out using the R-vine classifiers proposed in section 4.

6.2. Feature Extraction

We consider features based on the image gradient $g(x) \in \mathcal{R}^2$ computed at each pixel x of the image. The gradient vector is characterized by the magnitude $|g(x)|$ and phase $\phi(x)$. These features are appropriate to detect the patterns presented by sand dunes, since they describe the directional and periodic characteristics of the dunes [3]. In particular, the phase and magnitude histograms try to capture the typical edge structure of the local shape of a dune with a controlled degree of invariance to local geometric and radiometric factors.

The phase histogram associated to the k^{th} cell C^k is given by

$$h_i^k = \sum_{x \in C^k} b_i\left(\phi(x)\right)$$

where $b_i(\phi) = \begin{cases} 1, & \text{if } \phi \in i^{th} \text{ bin (of the histogram)} \\ 0, & \text{otherwise} \end{cases}$

The magnitude histogram associated to the k^{th} cell C^k is given by

$$\tilde{h}_i^k = \sum_{x \in C^k} \tilde{b}_i(|g(x)|)$$

where $\tilde{b}_i(|g|) = \begin{cases} 1, & \text{if } |g| \in i^{th} \text{ bin (of the histogram)} \\ 0, & \text{otherwise} \end{cases}$

The 180 features that were computed in our experiments for each image block (constituting nine cells) result from the following:

- 99 features for the magnitude, resulting from 11 bins per cell (considering four unit intervals between a minimum of 0 and a maximum of 40);
- 81 features for the phase, resulting from 9 bins per cell (considering an angular interval of 20°).

A normalization step is performed globally for each image and for each individual feature in order to have the features in $[0, 1]$.

6.3. Image Database

The image analysis is conducted on two databases with a total of 230 MOC-NA (Mars Orbiter Camera Narrow Angle) images: One database has 160 images representative of the major types of Martian dunes identified in [34] (see Fig. 4); and the other has 70 images containing other geomorphological structures that can be confounded with dunes, such as channels, crater rims, and textured terrain, among others.

Each original gray-scale image of the first database (see Fig. 6 (left)) has associated its ground-truth image, which was obtained by manually delineating the contours of the dunes therein contained, indicating the dune and not-dune regions (see Fig. 6 (center)).

In order to compare the ground-truth with the result produced by the classifier, both images had to be in the same format, so that the ground-truth images were tiled in the same fashion as the original ones (Fig. 6 (right)). In this image there are three types of cells: dune, in green; non-dune, in yellow; and unclassifiable, in grey. To assign one of these labels to a cell, the area of the block (from which the cells is the centre) that is occupied by ground-truth dune was computed: If this area is higher than 30% of the number of pixels of the block, the cell is considered dune (in green); if it is less than 10%, the cell is non-dune (in yellow); if the area is between 10 and 30%, the cell is not classified as any decision is considered ambiguous (in gray). In the 230 images, there are a total of 370019 cells, of which 112029 belong to the dune class and 257990 cells to the non-dune class. The ambiguous cells have been removed from the study.

7. Experiments

In this section we present and discuss the numerical results obtained through the experiments in order to test and evaluate the R-vine-based supervised approach applied to the dune detection problem. In the sections that follow, we first analyze several R-vine classifiers, which differ from each other in the number of trees, the families of selected pair-copulas and the types of univariate distributions used to model the gradient histogram features. We then analyze how the behavior of classifiers is affected when they use the same structure for the two classes.

As the approach presented in this work is described as an evolution of the D-vine-based classifiers introduced in [16], here we carry out experiments with both approaches. Finally, comparisons of R-vine classifiers and other state-of-the-art approaches in supervised classification are also provided for the Martian dune classification problem.

7.1. Experimental Framework

As explained in Section 6.1, the methodology for DCP is based on cells extracted from images of the surface of Mars. The prior probabilities of being a dune and non-dune cell are approximately 0.3 and 0.7 respectively.

The particular types of classifiers that we test in the numerical study are listed bellow. The same notation applies to the case of the classifiers based on D-vines, although we only refer to the R-vine classifiers for the sake of clarity. The number of trees in each class is determined with the truncation method presented in Algorithm 1-Step 6.

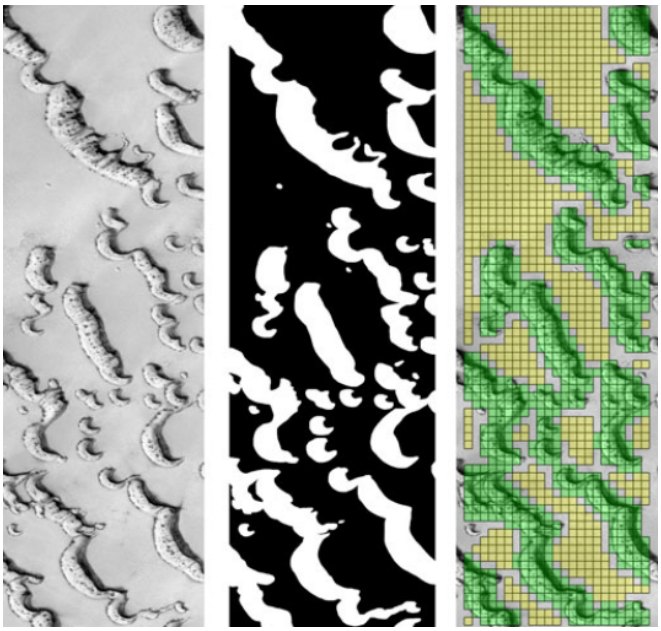


Figure 6: Image preparation: (right) original gray-scale image MOC-NA E18-00494; (center) manually drawn binary ground-truth; (right) tiling of the ground-truth in cells, where the dune cells are in green, non-dune cells are in yellow, and grey cells are unclassifiable. Image credits: NASA/JPL/MSSS. This figure is available in color online at wileyonlinelibrary.com/journal/espl.

- *Unmixed classifiers*
  - R-vine-P-g is the simplest R-vine classifier of this group. It comprises only product (P) copulas and Gaussian (g) margins. This classifier assumes that the variables are not correlated.
  - R-vine-t*-t*-N-g builds R-vines distributions where the first ' t*' indicates the number of trees in the dune class and the second 't*' indicates the number of trees in the non-dune class (the symbol '*' is the mask for an integer number ≥ 1 representing the number of trees); only normal copulas and Gaussian margins are used.
- *Partially-mixed classifiers*
  - R-vine-t*-t*-Sel-g builds R-vines distributions with t*-t* trees in the dune and non-dune class respectively; select different pair-copula families (Sel), whereas all the margins are Gaussian (g).
- *Fully-mixed classifiers*
  - R-vine-t*-t*-Sel-sel builds R-vines distributions with with t*-t* trees in the dune and non-dune class respectively; pair-copulas and margins can be of different types.

Within the group of candidate bivariate copulas, for the partially-mixed and fully-mixed classifiers we have included families that describe different forms of bivariate dependence structure in terms of symmetry, orientation and tail behavior they can represent. The product copula describes the independence. The normal and Student's t copulas model both positive and negative associations, they are symmetric and hence the lower and upper tail dependence coefficients are the same [12], the Student's t copula is lower and upper tail dependent while the normal is neither lower nor upper tail dependent. The Clayton and Gumbel copulas describe positive associations and are non-symmetric.Clayton is lower tail and Gumbel is upper tail dependent. The rotated by 90, 180 and 270 degrees versions of the Clayton and Gumbel copulas are included in the candidate copulas. The rotation by 90 and 270 degrees allows for the modeling of negative dependence, which is not possible with the non-rotated versions [12]. Obviously, there is a trade off between the flexibility that the inclusion of a wide variety of copulas provides, and the cost of learning vines selecting a large set of copula types. For the application addressed in this paper, we have assumed that a greater modeling capacity is better even at the expense of a higher computation cost. The acronym 'Sel' in the name of the classifiers denotes the case in which the classifier makes selection of the copula family.

Regarding the margin families, for the fully-mixed classifiers we use four parametric distributions: Gaussian, Student's t, Beta and Gamma. In addition to that, and in order to avoid making assumptions about the distribution of the data, we use the empirical distribution based on a Gaussian kernel. This kernel produce a smooth, continuous function) to represent the probability distribution using the sample data [10]. The bandwidth of the Gaussian kernel, which control the smoothness of the resulting density, is selected according to the rule-of-thumb bandwidth: $1.06\sigma/m^{1/5}$, with σ being the standard deviation of the Gaussian and m is the sample size. The acronym 'sel' in the name of the classifiers denotes the case in which the classifier makes selection of the margin family.

To evaluate the performance of the classifiers, we use as metrics the accuracy and area under the ROC curve (AUC) [11, 21, 49], which are estimated from 30 repetitions of a five-fold cross-validation [51]. Each fold has around 74003 cells of which around 22405 belong to the dune class and 51598 belong to the non-dune class.

To analyze the performance of the classifiers, we provide statistical comparisons using the Kruskal-Wallis and post-hoc Dunn statistical tests [18], which allow us to establish the statistical significance of the results according to AUC.

The R-vine classifiers and the experimental framework have been implemented in `R` language. The main packages that implement both vine-copula models and R-vine-based classification procedures used in this work are `copulaedas` [30, 29], `vines` [31], VinecopulaedasExtra [7], `rvclass` [8], and `VineCopula` [53].

7.2. Data Exploration

The assumption that underlies the application of R-vine classifiers to the DCP is that the univariate distributions of the features as well as the patterns of correlation among them are different in both classes. To test this assumption, we explore dune and non-dune instances through visual analytics tools.

To begin with, we analyze graphically the distributions of some of the variables in the two classes. Fig. 7 shows symmetric violin plots [35] of 6 (out of the 180) features arbitrarily chosen: X_81, X_84, and X_96 belong to the set of magnitude features and X_100, X_109, and X_153 belong to the set of phase features. The violin graph is a normal kernel density plot that is rotated and placed on each side to show the distribution shape of the data, where the vertical axis represents the values of x and the horizontal axis is the density of x. Values in the wider sections (in the bottom part) of the violin are more probable than those in the narrower sections. From these charts, we observe that none of these variables are normally distributed and that the shape of the distribution is different in each class: In X_100, X_109, X_153 and X_96, the non-dune class has wider sections than the dune class. However, in X_81 and X_84 the opposite occurs, being the dune class the one which has the widest sections; in addition, the widest section appears in opposite positions in each class –the highest probability is on the zero value in the dune class, and on the one value in the non-dune class. A final remark with regard to this topic, is that to properly model certain shapes such as those shown by the variables X_81 and X_96, it is convenient to include empirical marginals (smoothed with Gaussian kernel) in the group of distributions to be used in Algorithm 1, Step 1.

Next, we assess graphically the shape of the dependence in both classes. In Fig. 8, the copula data is transformed to have standard normal margins over the same six features used in the violin charts of Fig. 7. This figure shows scatter plots along with pairwise Pearson's correlation values above and contour plots with standard normal margins below the diagonal. The bivariate contour plots differ from dispersed and elliptical shapes (which characterize the pairwise independence and linear dependence, respectively) showing the need to non-Gaussian copulas. In addition to that, this figure also reveals that both the strength and the types of dependence for the same pair are quite different between the two classes.

In summary, the graphical exploratory data analysis performed in this section shows that both the marginal behavior of the gradient features as well as the interactions among them are different for dune and non-dune sample data. These findings further justify the use of the R-vine-based approach to design probabilistic classifiers.

7.3. Analysis of R-vine Classifiers

In the previous section, we have presented evidence that dune and non-dune data are different to each other in terms of the statistical characteristics of magnitude and phase histogram features as well as the pairwise correlations between them. This evidence supports the main working hypothesis of the experiments of this section: We believe

[7] `VinecopulaedasExtra`, https://github.com/DianaCarrera, https://github.com/DianaCarrera/VinecopulaedasExtra
[8] `rvclass`, https://github.com/DianaCarrera, https://github.com/DianaCarrera/rvclass

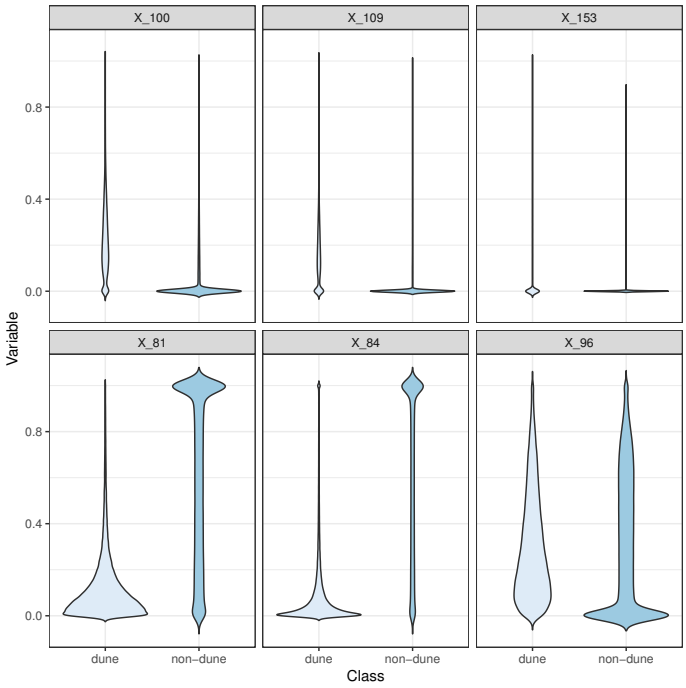


Figure 7: Symmetric violin plots with the distribution of 6 (of the 180) features arbitrarily chosen using the data of the respective class.

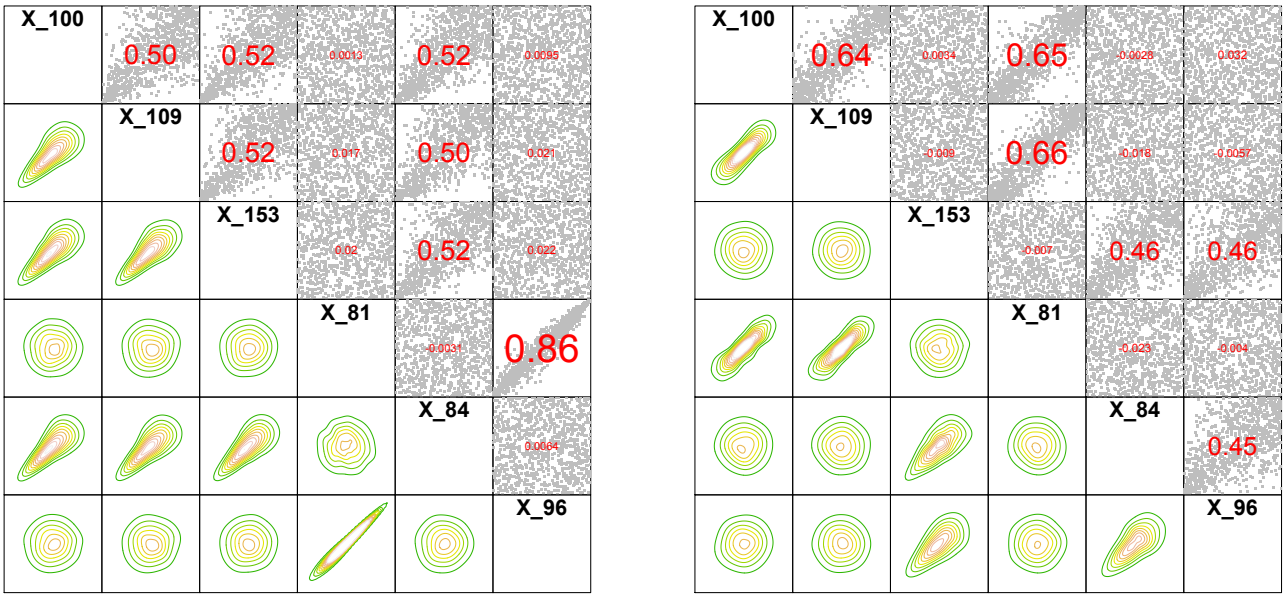


Figure 8: Pairs plot over 6 (out of the 180) features arbitrarily chosen from the dune (left panel) and non-dune sample data (right panel) with scatter plots above and contour plots with standard normal margins below the diagonal.

that the better the R-vine distribution of each class captures the distribution of the variables as well as the pairwise interactions between them, the better the predictive ability of R-vine classifiers.

To test the working hypothesis, we compare the different types of D- and R-vine classifiers (presented in Section 7.1), which differ among themselves in terms of the number of trees, the pair-copula families, and the types of univariate margins that each classifier encodes. Accuracy and AUC results obtained with unmixed, partially-mixed and fully-mixed classifiers in the DCP can be appreciated in Table 2.

Let us start analyzing the performance of the four unmixed R-vine classifiers. They have in common that all margins are Gaussian, whereas they differ in the number of trees and types of copulas they use: product or normal. On one hand, the results reveal that the two classifiers that select normal copulas: D-vine-t2-t1-N-g and R-vine-t1-t1-N-g are more accurate than the versions that only uses product copulas: D-vine-P-g and R-vine-P-g, which are totally unaware of the correlation structure of the problem, exhibiting the poorest accuracy ($81,0\%$ and $84,3\%$ respectively) and AUC ($76,1\%$ and $88,2\%$ respectively). On the other hand, among the two unmixed classifiers with normal copulas, R-vine-t1-t1-N-g, which has one tree in each class, reaches higher accuracy and AUC values than D-vine-t2-t1-N-g with two trees in the first class and one tree in the second class. In general, the results obtained suggest the existence of pairwise correlations between the variables and that capturing these correlations by an R-vine distribution can lead to improve the behavior of the classifier. This remark is confirmed when we analyze the four classifiers that combine copulas from different families (partially-mixed and fully-mixed), which achieve higher accuracy and AUC values than those that select only normal ones. This finding suggests the presence of some more complex interactions (as asymmetric dependence structures) that are better modeled by the different versions of Gumbel and Clayton copulas considered in our experiments.

Now, let us analyze the effect of marginal distributions in improving the functioning of classifiers presented in Table 2. We can appreciate that fully-mixed classifiers that encode Beta, Gamma and Gaussian margins outperform the partially-mixed classifiers that assume that all the variables are Gaussian. We explain this result through the variable X_84 , whose shape is similar to other variables of DCP. In Fig. 9, we have plotted the Beta, Gamma and normal density curves estimated from the sample data of this variable in order to visually check how these curves resemble the smoothed empirical curve for both classes respectively. For the dune class (left panel), we can appreciate that the Gamma and smoothed empirical curves overlap each other so that they are almost indistinguishable. in the figure, whereas the Beta and Gaussian distributions provide a very poor fit. For the non-dune class (right panel), the best fit is achieved with the Beta distribution, its curve is the one that resembles the most the smoothed empirical curve, whereas the Gamma and Gaussian distributions produce a poor fit.

Summarizing this section, we can say that from all tested classifiers, R-vine-t5-t7-Sel-sel is the most accurate (with an accuracy of $95,2\%$ and an AUC of $98,8\%$) and, at the same time, the most flexible. A look to the differences in the frequencies of each type of margins and of copula families between the R-vines learned for each class of R-vine-t5-t7-Sel-sel provides an insight into how the different data distributions are captured by the R-vine models:

- For the dune class, it chooses (in ascending order) 9 Student' t, 10 Gaussian, 38 smoothed empirical, 48 Beta, and 75 Gamma margins; and 98 Rotated Clayton 90°, 102 Rotated Clayton 270°, 109 Gumbel, 117 Student's t, 135 normal, 151 product, and 173 Clayton, pair-copulas, and copulas (there are 885 edges in an R-vine with 5 trees and 180 variables).

- For the non-dune class, it chooses (in ascending order) 12 Student' t, 13 Gaussian, 28 smoothed empirical, 46 Gamma, and 81 Beta margins; and 96 Rotated Clayton 270°, 113 Rotated Clayton 90°, 184 product, 197 Gumbel, 199 Student's t, 202 normal, and 241 Clayton pair-copulas (there are 1232 edges in an R-vine with 7 trees and 180 variables).

From here on, the next experiments are performed only for partially-mixed and fully-mixed D-vine and R-vine classifiers.

7.4. Using a Common R-vine Structure

The proposed R-vine-based classification strategy requires that the learned R-vine distributions of the classes be different each other in order to the classifier can distinguish which class an instance belongs to. When using CS1 and CS2 methods, the linked variables are the same in both classes but not the pair-copulas, which are selected and estimated from the corresponding data. We believe that these methods make easier to interpret and identify which pairwise dependencies are those that contribute to characterize the classes and to assess how the pairwise dependencies changes among them.

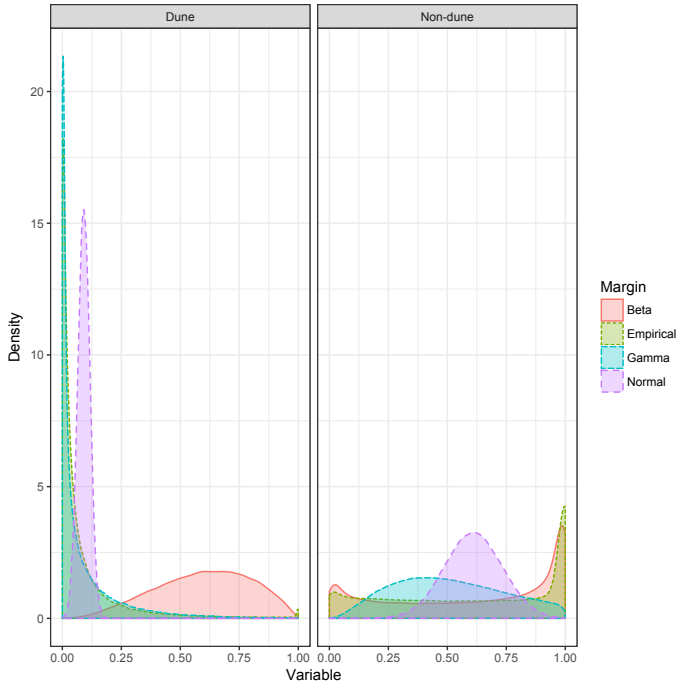


Figure 9: Comparison of Beta, Gamma and Gaussian margins with the empirical distribution for the same variable in the dune and non-dune classes.

Table 2: Classification accuracy and AUC results in the DCP with unmixed, partially-mixed and fully-mixed R-vine classifiers learned with the DS method.

Classifier	Accuracy	AUC
DS method, unmixed classifiers		
D-vine-P-g	$81,0$	$76,1$
R-vine-P-g	$84,3$	$88,2$
D-vine-t2-t1-N-g	$85,7$	$86,5$
R-vine-t1-t1-N-g	$87,3$	$90,1$
DS method, partially-mixed classifiers		
D-vine-t4-t2-Sel-g	$89,8$	$91,3$
R-vine-t3-t4-Sel-g	$92,6$	$94,0$
DS method, fully-mixed classifiers		
D-vine-t4-t4-Sel-sel	$92,1$	$92,5$
R-vine-t5-t7-Sel-sel	$95,2$	$98,8$

The price to be paid for learning an R-vine tree-structure per class is that the models can not be directly compared in terms of the selected pair-copulas because the variables that determine the most important interactions may be different in each class. A classifier with shared structure is more amenable for identification of the differences between classes. That said, here we investigate how the performance of R-vine classifiers is affected if only a single tree-structure is estimated for all classes. In our approach, only the tree-structure is common for the R-vine distributions of both classes, whereas the pair-copulas and their parameters are selected individually from data of the corresponding class.

Let us begin with the discussion of the results. Table 3 presents the accuracy and AUC statistics for the partially-mixed and fully-mixed classifiers learned with the CS1 and CS2 methods (we use the cross-validation methodology presented in Section 7.1). We can appreciate that these results are coherent with those presented in Section 7.3: The classifiers that combine different types of pair-copulas and margins achieve higher accuracies. However, the most interesting result comes from comparing DS-based classifiers (see Table 2) with those learned using CS1 and CS2. Comparing classifiers with the best performance learned with DS, CS1 and CS2, we can see that the accuracy and AUC of the DS-based R-vine-t5-t7-Sel-sel is 95.2% and $98,8\%$, whereas with CS1-based R-vine-t3-t3-Sel-sel the accuracy decreases around $4,2\%$ and the AUC decreases around $6,4\%$, and with CS2-based R-vine-t4-t3-Sel-sel the accuracy decreases around $3,6\%$ and the AUC decreases around $5,6\%$.

Figures 10 - 12 account for such behavior through the most accurate R-vine classifiers learned with the three learned strategies methods. Fig. 10 displays box plots of Kendall taus associated to the edges of the first tree learned with CS1 (left), CS2 (center) and DS (right), respectively, for the dune and non-dune classes. Each box plot is drawn from 179 values of Kendall's tau (there are 179 edges in the first tree of an R-vine of 180 variables). It is important to clarify that in the cases of CS1 and CS2 box plots, the Kendall's tau values are estimated for the edges of the common tree-structure, using, however, the copula data of the respective class. From this figure, we see that the absolute maximum, the third quartile, the median, the first quartile and the minimum of the dune and non-dune box plots for the DS reached higher absolute Kendall's tau values than in the other four box-plots. It is easy to see that R-vine classifiers that learn the tree-structure of each class can more freely accommodate the strongest dependencies than those that use a common tree-structure.

Fig. 11 confirms this clear trend in favor of DS method. The x-axis represents the edges belonging to the first tree learned with CS1 (circle), CS2 (triangle) and DS (square). The edges are arranged in descending order according to the absolute Kendall's tau values computed from the copula data of the dune class; the y-axis represents the Kendall's tau value associated to the edges in the x-axis. This figure is made only for the dune class since for the non-dune class the behavior is similar. The DS method has more freedom than CS1 and CS2 to include a greater number of strong dependencies. However, with CS1 and CS2, the strong edges for one class may be out of the tree since they are weak edges for the other class. The restriction of common structure together with that of being a tree prevents the insertion of strong edges that are replaced by weak ones. This behavior leads to an increase in the number of product copulas fitted, as can be seen in the stack graph of Figure 12, which shows the bivariate copula families selected by each method in the first tree for both classes. The DS method selects the smallest number of product copulas and the largest number of Clayton copulas for both classes.

The use of a single tree-structure can facilitate the interpretation of R-vine classifiers since the set of dependencies explicitly represented is the same for all classes. This means that it is possible to compare, for each edge of the tree, which copula families are learned for this edge for the two classes. When the copula family coincides in the two trees, the strength of the dependence can help to characterize and interpret the differences between classes. In Figure 13 we show the copula families assigned to the 20 strongest edges of the first tree found by CS1, CS2, and DS. On one hand, we can see that the classifier that DS-based selects a large number of Clayton copulas. In fact, in these edges most of the selected copulas belong to this family (9 in each class), and only 3 and 5 normal copulas are selected in the dune and non-dune classes respectively. Conversely, the classifiers that share a common tree-structure select more normal copulas (13 and 12 with CS1, and 8 and 9 with CS2) than the classifier that use different tree-structures being at the same time the most accurate.

In summary, although the constraint of a common tree-structure can limit the flexibility of the model and therefore impact the accuracy of the classifiers, the experiments show that the impact is not severe, which could be explained by the fact that R-vine classifiers with shared tree-structure still keep a high degree of flexibility thanks to the use of copulas from different families.

Table 3: Classification accuracy and AUC results in the DCP with partially-mixed and fully-mixed R-vine classifiers using the methods CS1 and CS2.

Classifier	Accuracy	AUC
CS1 method, partially-mixed classifiers		
D-vine-t2-t2-Sel-g	$85,6$	$90,2$
R-vine-t2-t3-Sel-g	$88,4$	$91,3$
CS1 method, fully-mixed classifiers		
D-vine-t3-t2-Sel-sel	$89,9$	$87,2$
R-vine-t3-t3-Sel-sel	$91,2$	$92,4$
CS2 method, partially-mixed classifiers		
D-vine-t3-t2-Sel-g	$87,3$	$89,3$
R-vine-t3-t2-Sel-g	$90,8$	$89,2$
CS2 method, fully-mixed classifiers		
D-vine-t3-t3-Sel-sel	$89,2$	$89,8$
R-vine-t4-t3-Sel-sel	$91,7$	$93,2$

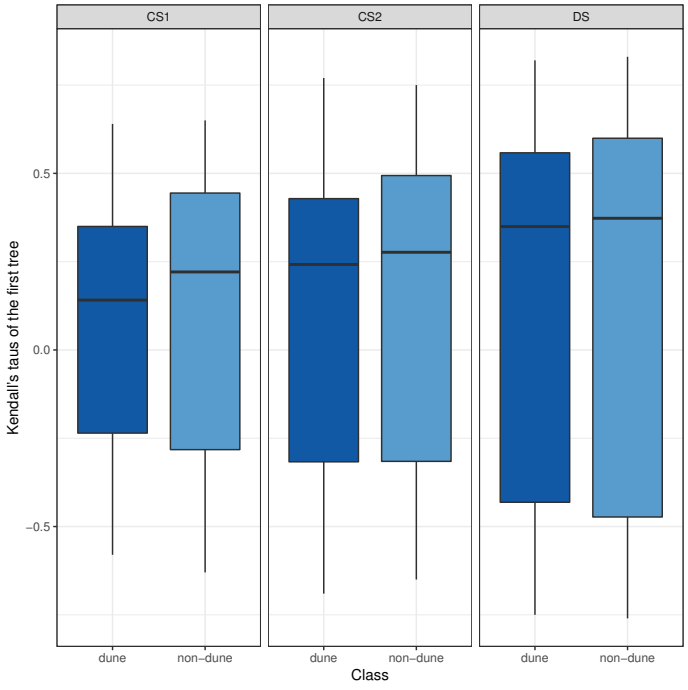


Figure 10: Box plots of empirical Kendall taus associated to the edges of the first tree learned with the CS1 (left), CS2 (center) and DS (right) methods for the dune (dark blue) and non-dune classes (light blue).

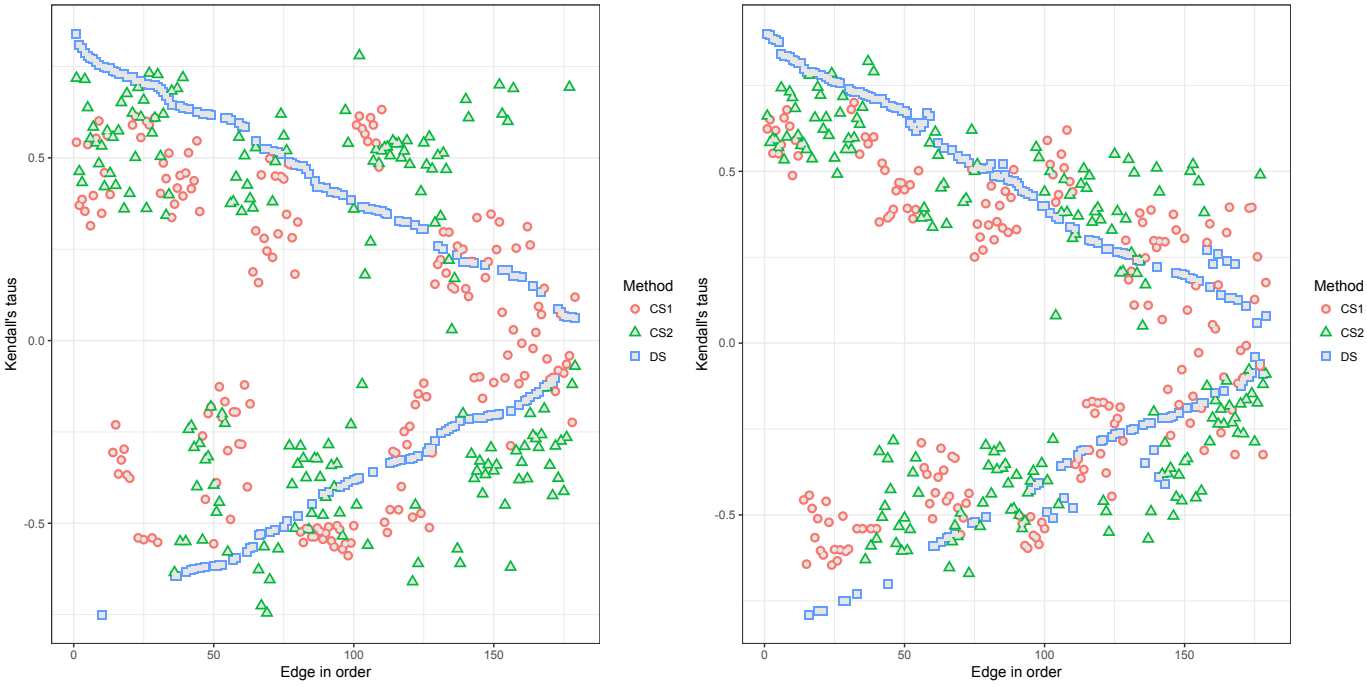


Figure 11: Scatter plots of the empirical Kendall's tau values associated to the 179 edges of the first tree built with the CS1 (red), CS2 (green) and DS (blue) methods for the dune class (left panel) and the non-dune class (right panel). In the x-axis, the edges are arranged in descending order according to the absolute empirical Kendall's tau values calculated from the copula data of the corresponding class.

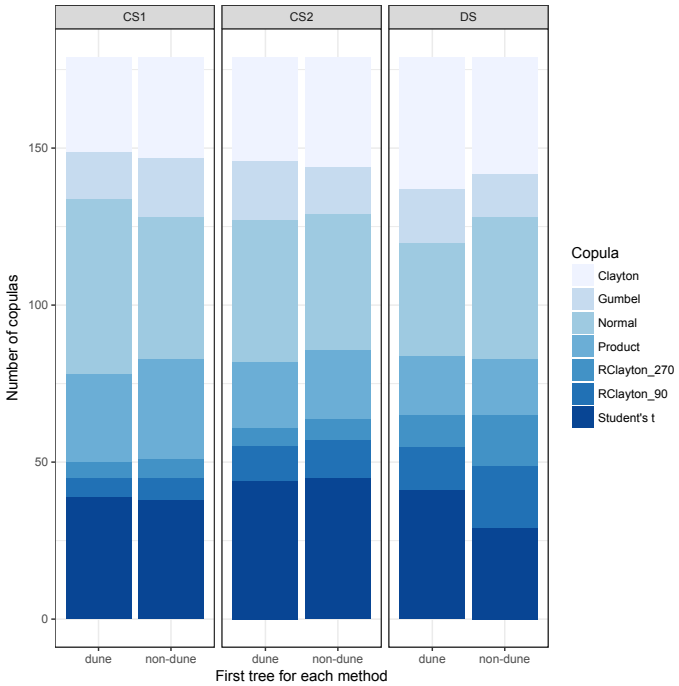


Figure 12: Number of product, normal, Student's t, Clayton, Gumbel, and rotated (by 90°, 180° and 270°) Clayton and Gumbel copulas fitted in the first tree built with the CS1, CS2, and DS methods for the dune and non-dune classes.

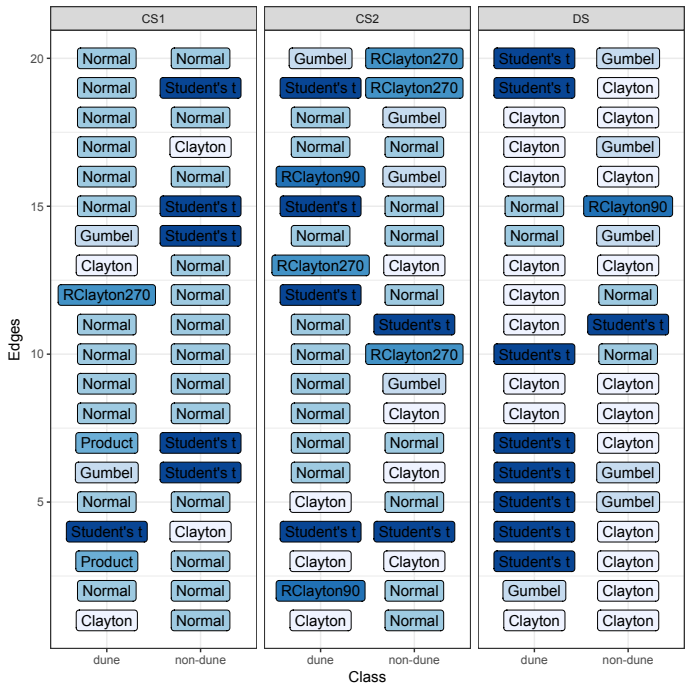


Figure 13: Types of copulas fitted in the 20 strongest edges of the first tree built with the CS1, CS2, and DS methods for the dune and non-dune classes.

7.5. Comparing R-vine Classifiers with Other Classification Approaches

In this section, we assess the performance of R-vine classifiers in comparison with 10 state-of-the-art algorithms from scikit-learn [48], a widely used machine learning library in Python. Each algorithm and its hyperparameters are described in Table 4. They cover commonly applied approaches to classification tasks including linear and non-linear classifiers, tree-based classifiers, ensemble classifiers, and distance-based classifiers. Some of these algorithms are able to capture non-linear interactions between the variables, while others incorporate regularization techniques. For more information on these algorithms, see [33]. For the optimization of the hyperparameters of each classifier, we first split the set of hyperparameters in two groups, those that have a strong influence on the results of the algorithm and those with a low relevance according to the suggestions given in [47]. Then, for each algorithm we perform grid search to optimize the first group of hyperparameters using the cross-validation methodology presented in Section 7.1. The second group of hyperparameters (those with low relevance) were set to the default values in the scikit-learn implementation of the algorithms. The optimized hyperparameters and the corresponding best values obtained via grid search are shown in Table 4.

In this numerical comparison, we focus on the most flexible variants of R-vine classifiers built with the DS, CS1, CS2 methods: namely R-vine-t5-t7-Sel-sel (with DS), R-vine-t3-t3-Sel-sel (with CS1), and R-vine-t4-Sel-sel (with CS2). We renamed them, in short, R-vine-DS, R-vine-CS1, and R-vine-CS2 respectively. The same notation applies to D-vine-t3-t3-Sel-sel or D-vine-DS in short.

Numerical comparisons according to the accuracy and AUC are given in Table 5 (in increasing order according to AUC). In order to assess the statistical significance of the observed differences in algorithm performance, we use the Kruskal-Wallis statistical test on the AUC's values to determine whether all the groups originate from the same distribution. If the null hypothesis is rejected ($p\text{-}value < 0.05$), a post-hoc test is applied to all the sample data pairs, looking for differences between them. The Dunn test is used for the pairwise comparison. The results of the tests for the 14 algorithms are shown graphically in Figure 14. Vertical lines stand for the algorithms. Horizontal lines mean that there are no statistical significant differences among algorithms that cut. On the contrary, the differences among algorithms are statistically significant if there is no horizontal line that cuts the vertical algorithm line.

Among all compared algorithms, R-vine-DS reaches the highest accuracy ($96,4\%$) and AUC ($98,8\%$), which is a remarkable performance. It is followed closely, first, by the other classifiers based on vine-copula models: R-vine-CS2, D-vine-DS, and R-vine-CS1. The most important fact to highlight here is that the classifiers based on

Table 4: Classification algorithms, the associated optimized hyperparameters and their best values obtained via grid search. Both the method names (in the first column) and the hyperparameter names (in the second column) correspond to their implementations in scikit-learn library.

Algorithms	Hyperparameters	Best Values
GB - Gradient Boosting [25]	`n_estimators`: Number of decisions trees.	500
`GradientBoostingClassifier()`	`learning_rate`: Shrinks the contribution of each tree.loss	0.1
	`loss`: Loss function to be optimized.	`'deviance'`
	`max_depth`: Maximum depth of the individual trees.	11
	`max_features`: Number of features to consider when looking for the best split.	`'log2'`
RF - Random Forest	`n_estimators`: Number of trees in the forest.	500
`RandomForestClassifier()`	`criterion`: Function to measure the quality of a split.	`'entropy'`
	`max_depth`: Maximum depth of the individual trees.	11
	`max_features`: Number of features to consider when looking for the best split.	`'auto'`
SVM - Support Vector Machines	`C`: Penalty parameter of the error term (regularization).	1.0
`SVM()`	`kernel`: Kernel type	`'rbf'`
	`gamma`: Kernel coefficient for the `'rbf'`, `'poly'` and `'sigmoid'` kernels.	0.1
	`degree`: Degree for the `'poly'` kernel.	3
	`coef0`: Independent term for the `'poly'` and `'sigmoid'` kernels.	10
LR - Logistic Regression [64]	`penalty`: L1 or L2 penalization.	`'l2'`
`LogisticRegression()`	`C`: Loss function to be optimizad.	1.5
	`fit_intercept`: If the intercept is added to the decision function.	`True`
LDA - Linear Discriminant Analysis [24]	`solver`: Solver to use.	`'svd'`
`LinearDiscriminantAnalysis()`	`shrinkage`: Shrinkage parameter.	`None`
EDT - Extra Decision Trees	`n_estimators`: Number of trees in the forest.	1000
`ExtraTreesClassifier()`	`criterion`: Function to measure the quality of a split.	'entropy'
	`max_depth`: Maximun depth of the individual trees.	11
	`max_features`: Number of features to consider when looking for the best split.	'log2'
KNN - K Nearest Neighbors	`n_neighbors`: Number of neighbors to use.	15
`KNeighborsClassifier()`	`weights`: Function to weight the neighbors' votes.	`'distance'`
NN - Multilayer Perceptron Neural Network	`hidden_layer_sizes`: The i^{th} element of the tuple represents the number of neurons in the i^{th} hidden layer.	$(50,)$
`MLPClassifier()`	`activation`: Activation function for the hidden layer.	'logistic'
GNB - Gaussian Naive Bayes.	No parameters.	
`GaussianNB()`		
DT - Decision Tree	`criterion`: Function to measure the quality of a split.	'entropy'
`DecisionTreeClassifier()`	`max_depth`: Maximun depth of the tree.	11
	`max_features`: Number of features to consider when looking for the best split.	'none'

Table 5: Numerical comparisons of R-vine classifiers with other approaches with respect to the accuracy and AUC in the DCP. The compared R-vine classifiers belong to the fully-mixed group and are learned with the DS, CS1 and CS2 methods. Classifiers are ranked in descending order according to AUC to AUC in the same order as they appear in Figure 14.

Rank	Classifier	Accuracy	AUC	Rank	Classifier	Accuracy	AUC
1	R-vine-DS	$95,2$	$98,8$	8	LR	$90,2$	$90,1$
2	R-vine-CS2	$92,1$	$92,5$	9	LDA	$90,4$	$89,8$
3	D-vine-DS	$91,7$	$93,2$	10	EDT	$90,6$	$89,2$
4	R-vine-CS1	$91,2$	$92,4$	11	KNN	89.4	$88,3$
5	GB	$90,7$	$91,7$	12	NN	$88,7$	$85,3$
6	RF	$90,0$	$91,0$	13	GNB	$87,8$	$81,2$
7	SVM	$89,0$	$90,2$	14	DT	$89,2$	$79,1$

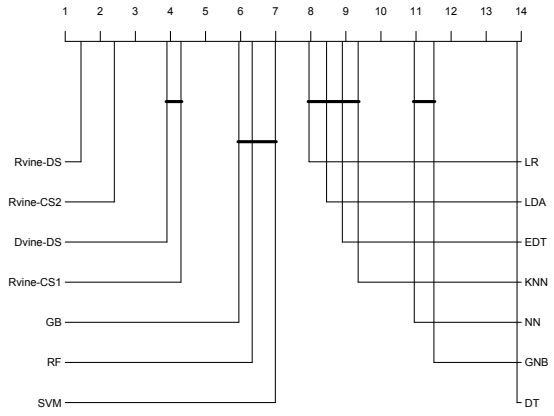


Figure 14: Statistical comparisons of tested classification approaches based on the Kruskal-Wallis and post-hoc Dunn statistical tests according to AUC.

vine-copula models are the ones that obtain the highest AUC, being these results statistically significant in relation to the particular instances of algorithms tested in the DCP. It is also remarkable that classifiers based on common structure strategies CS1 and CS2 are in the top positions.

A final remark is that these results confirm that the methodology based on gradient histograms, combined with machine learning algorithms, is a good approach to deal with the DCP. These features adequately describe the characteristics of the dune and non-dune images, allowing the algorithms to discriminate between the two classes.

8. Conclusions

This paper proposes the use of probabilistic classifiers based on R-vines distributions to address the dune detection problem in a supervised classification approach. The general scheme learns an R-vine distribution per class (dune and non-dune). To classify a given observation, the probability given by the R-vine of each class is computed, and the observation is assigned the class with the highest probability. This approach is successfully applied to the DCP where complex dependence structures usually arise. R-vines include as a particular case D-

vines, which have been previously applied to implement classification approaches. Therefore, R-vine classifiers extend the representation capabilities and expressiveness of D-vine classifiers.

The numerical results show that the better the R-vines distributions represent the marginal behavior of the variables and the interactions between them, the more accurate the classification is. Among all tested R-vine classifiers, the most flexible is, at the same time, the most accurate: R-vine-t5-t7-Sel-sel using the DS method. It utilizes a hierarchical structure (with five trees in the dune class and seven trees in the non-dune class) to represent conditional dependencies of a high order. It also combines different pair-copula families in the same decomposition, and uses different types of univariate distributions to model the selected features.

The methods CS1 and CS2 proposed here allow the building of the same tree-structure for the dune and non-dune classes. Although this strategy prevents the insertion of several strong dependencies, the classifiers are able to achieve high accuracy and AUC values thanks to the use of copulas and margins from different families. For the DCP, the results of these classifiers outperform traditionally applied classifiers illustrating that while incorporating constraints in their structure, they are still competitive. Besides, it can be anticipated that the advantages of these strategies should be more evident in problems with multiple classes ($K > 2$). In each R-vine tree-level, the algorithm only has to build one MWST instead of K MWSTs.

Ongoing topics demanding future work are the following:

- One relevant question is how to modify the learning methods in order to increase the interpretability of the problem, while keeping or even improving the accuracy. The rationale behind CS1 and CS2 is that we can determine and easily evaluate how the R-vine distributions with the same structure change the types of copulas and their parameters when adapting to class distributions. Other strategies for finding a good MWST that is common for all the classes should be considered.
- We have modeled pairwise dependencies with parametric bivariate copulas that describe a wide range of dependence structures of a sample data. It is worth testing (smoothed) empirical copulas to model complex forms of dependence that can not be captured by any parametric copula. We believe that a more accurate shape for the bivariate data may lead to increasing the predictive ability of R-vine classifiers.
- Investigating MLE approaches for copula selection, and its combination with strategies that learn a single R-vine tree-structure for all classes.
- Investigating common structure-based strategies in multi-class classification problems.

Acknowledgements

This work is partially supported by the Basque Government (IT609-13 and Elkartek), and Spanish Ministry of Science and Innovation (TIN2016-78365-R). Jose A. Lozano is also supported by BERC 2014-2017 and Elkartek programs (Basque government) and Severo Ochoa Program SEV-2013-0323 (Spanish Ministry of Economy and Competitiveness)

Appendix

We learn an R-vine distribution of four random variables $\mathbf{X} = (X_1, X_2, X_3, X_4)$, according to Algorithm 1 as follows:

1. Estimate the univariate cumulative and density functions $F_i(X_i)$ and $f(X_i)$ from the original data $\mathbf{D_X}$ defined over $\mathbf{X}$, where
$$\begin{array}{rcl} X_1 & \sim & F_1, \\ X_2 & \sim & F_2, \\ X_3 & \sim & F_3, \quad \text{and} \\ X_4 & \sim & F_4. \end{array}$$
2. Obtain unconditional copula data $\mathbf{D_U}$ by evaluating the unconditional distribution functions $F_i(X_i)$, estimated in the previous step, i.e.,
$$\begin{array}{rcl} u_1 & := & F_1(x_1), \\ u_2 & := & F_2(x_2), \\ u_3 & := & F_3(x_3), \quad \text{and} \\ u_4 & := & F_4(x_4). \end{array}$$

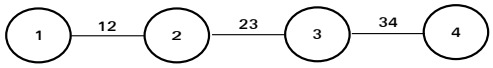
3. Compute Kendall's tau for each pair of variables $(1-2,\ 1-3,\ 1-4,\ 2-3,\ 2-4,\ 3-4)$ from the copula data $\mathbf{D_U}$ defined over $\mathbf{U} = (U_1, U_2, U_3, U_4)$ obtained in the previous step,

$$\begin{array}{c|cccc} & u_1 & u_2 & u_3 & u_4 \\ u_1 & & 0,8 & 0,4 & 0,2 \\ u_2 & & & 0,7 & 0,5 \\ u_3 & & & & 0,6 \\ u_4 & & & & \end{array}.$$

4. Build the T_1 (MWST) that maximizes the sum of the absolute Kendall's taus. Furthermore, select (unconditional) pair-copulas using the distance measure (17), and compute the parameters of the selected pair-copulas using the relationship between Kendall's tau and the dependence parameter of the corresponding bivariate copula (Table 1). Three unconditional pair-copulas

$$\begin{array}{l} c_{12}\left(u_1, u_2\right), \\ c_{23}\left(u_2, u_3\right), \text{ and} \\ c_{34}\left(u_3, u_4\right) \end{array}$$

are associated to the edges of the first tree as shown in the following representation:



5. Obtain conditional copula data $\mathbf{D_U}$ (needed for T_2) by evaluating conditional distribution functions using the bivariate copulas selected in T_1,

$$\begin{array}{rcccl} F\left(x_1 \mid x_2\right) & = & \frac{\partial C_{12}(F(x_1),F(x_2))}{\partial F(x_2)} & = & \frac{\partial C_{12}(u_1,u_2)}{\partial u_2}, \\ F\left(x_2 \mid x_3\right) & = & \frac{\partial C_{23}(F(x_2),F(x_3))}{\partial F(x_3)} & = & \frac{\partial C_{23}(u_2,u_3)}{\partial u_3}, \quad \text{and} \\ F\left(x_3 \mid x_4\right) & = & \frac{\partial C_{34}(F(x_3),F(x_4))}{\partial F(x_4)} & = & \frac{\partial C_{34}(u_3,u_4)}{\partial u_4}. \end{array}$$

The derivation of these expressions for the normal, Clayton and Gumbel copulas can be found in [2].

6. Compute the Kendall's tau for all possible pairs of variables $(12-23$ and $23-34)$ from the conditional copula data $\mathbf{D_U}$, defined over $\mathbf{U} = \left(F\left(X_1 \mid X_2\right), F\left(X_2 \mid X_3\right), F\left(X_3 \mid X_4\right)\right)$,

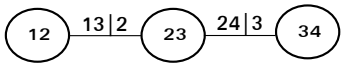
$$\begin{array}{c|ccc} & F\left(x_1 \mid x_2\right) & F\left(x_2 \mid x_3\right) & F\left(x_3 \mid x_4\right) \\ F\left(x_1 \mid x_2\right) & & 0.6 & \text{NA} \\ F\left(x_2 \mid x_3\right) & & & 0.5 \\ F\left(x_3 \mid x_4\right) & & & \end{array},$$

where NA means that the Kendall's tau is not being computed (the linking nodes of the edges $12-34$ do not share a common node).

7. Build the tree T_2 that maximizes the sum of absolute Kendall's taus and fit two conditional pair-copulas

$$\begin{array}{l} c_{13|2}\left(F\left(x_1 \mid x_2\right), F\left(x_3 \mid x_2\right)\right), \text{ and} \\ c_{24|3}\left(F\left(x_2 \mid x_3\right), F\left(x_3 \mid x_4\right)\right) \end{array}$$

that are assigned to the edges of the second tree; see the following representation:



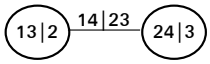
8. Obtain conditional copula data $\mathbf{D}_{\mathrm{U}}$ (needed for T_3) by evaluating conditional distribution functions using the bivariate copulas selected in T_2,

$$\begin{array}{rcl} F\left(x_1 \mid x_2 x_3\right) & = & \frac{\partial C_{13|2}(F(x_1|x_2),F(x_3|x_2))}{\partial F(x_3|x_2)}, \quad \text{and} \\ F\left(x_2 \mid x_3 x_4\right) & = & \frac{\partial C_{24|3}(F(x_2|x_3),F(x_3|x_4))}{\partial F(x_3|x_4)}. \end{array}$$

9. Build the tree T_3 and fit the conditional pair-copula

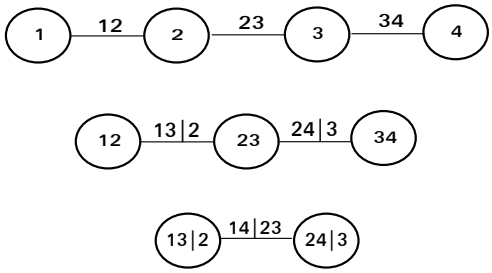
$$c_{14|23}\left(F\left(x_1 \mid x_2 x_3\right), F\left(x_2 \mid x_3 x_4\right)\right),$$

which is assigned to the edge on the last tree as shown in the figure below:



1. The result is an R-vine copula with three trees and five pair-copulas. That is,

$$f\left(x_1, x_2, x_3, x_4\right) = c_{12} \cdot c_{23} \cdot c_{34} \cdot c_{13|2} \cdot c_{24|3} \cdot c_{14|23} \cdot \prod_{i=1}^{4} f_i\left(x_i\right)$$



[1] Kjersti Aas. Modelling the dependence structure of financial assets: A survey of four copulas. Note SAMBA/22/04, Norwegian Computing Center, NR, Norway, 2004.

[2] Kjersti Aas, Claudia Czado, Arnoldo Frigessi, and Henrik Bakken. Pair-copula constructions of multiple dependence. *Insurance: Mathematics and Economics*, 44(2):182–198, 2009.

[3] Lourenço Bandeira, Jorge S. Marques, Jose Saraiva, and Pedro Pina. Automated detection of Martian dune fields. *IEEE Geoscience and Remote Sensing Letters*, 8(4):626–630, 2011.

[4] Lourenço Bandeira, Jorge S. Marques, Jose Saraiva, and Pedro Pina. Advances in automated detection of sand dunes on Mars. *Earth Surface Processes and Landforms*, 38(3):275–283, 2013.

[5] T. Bedford and R. M. Cooke. Probability density decomposition for conditionally dependent random variables modeled by vines. *Annals of Mathematics and Artificial Intelligence*, 32(1):245–268, 2001.

[6] T. Bedford and R. M. Cooke. Vines – a vew graphical model for dependent random variables. *The Annals of Statistics*, 30(4):1031–1068, 2002.

[7] N. Belgorodski. Selecting pair-copula families for regular vines with application to the multivariate analysis of European stock market indices, 2010.

[8] Mark A. Bishop. Nearest neighbor analysis of mega-barchanoid dunes, ar rub'al khali, sand sea: The application of geographical indices to the understanding of dune field self-organization, maturity and environmental change. *Geomorphology*, 120(3):186–194, 2010.

[9] Mary C. Bourke, Nick Lancaster, Lori K. Fenton, Eric J. R. Parteli, James R. Zimbelman, and Jani Radebaugh. Extraterrestrial dunes: An introduction to the special issue on planetary dune systems. *Geomorphology*, 121(1):1–14, 2010.

[10] A. W. Bowman and A. Azzalini. *Applied Smoothing Techniques for Data Analysis*. New York: Oxford University Press Inc, 1997.

[11] A. P. Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, 1997.

[12] E. C. Brechmann and U. Schepsmeier. Modeling dependence with C- and D-Vine copulas: The R package CDVine. *Journal of Statistical Software*, 52(3):1–27, 2013.

[13] Eike C. Brechmann, Claudia Czado, and Kjersti Aas. Truncated regular vines in high dimensions with application to financial data. *Canadian Journal of Statistics*, 40(1):68–85, 2012.

[14] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[15] D. Carrera. Modelado de dependencias con vines basados en cópulas bernstein, 2012. In Spanish.

[16] D. Carrera, R. Santana, and J. A. Lozano. Vine copula classifiers for the mind reading problem. *Progress in Artificial Intelligence*, 5(4):289–305, 2016.

[17] Yuhui Chen. A copula-based supervised learning classification for continous and discrete data. *Journal of Data Science*, 14(4):769–782, 2016.

[18] G. Corder and D. Foreman. *Nonparametric Statistics: A Step-by-Step Approach*. Wiley & Son, 2009. 2nd Edition.

[19] C. Czado. Pair-copula constructions of multivariate copulas. In P. Jaworski, F. Durante, W. K. Härdle, and T. Rychlik, editors, *Copula Theory and Its Applications*, volume 198 of *Lecture Notes in Statistics*, pages 93–109. Springer, Berlin, Heidelberg, 2010.

[20] Jeffrey Dissmann, Eike C. Brechmann, Claudia Czado, and Dorota Kurowicka. Selecting and estimating regular vine copulae and application to financial returns. *Computational Statistics & Data Analysis*, 59:52–69, 2013.

[21] T. Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, 2010.

[22] Lori K. Fenton and Rosalyn K. Hayward. Southern high latitude dune fields on Mars: Morphology, aeolian inactivity, and climate change. *Geomorphology*, 121(1):98–121, 2010.

[23] Lori K. Fenton, Anthony D. Toigo, and Mark I. Richardson. Aeolian processes in Proctor crater on Mars: Mesoscale modeling of dune-forming winds. *Journal of Geophysical Research: Planets*, 110(E6), 2005.

[24] Ronald A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of human genetics*, 7(2):179–188, 1936.

[25] Jerome H. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[26] N. Friedman and M. Goldszmidt. Building classifiers using Bayesian networks. In *National Conference on Artificial Intelligence*, pages 1277–1284, 1996.

[27] C. Genest and A. C. Favre. Everything you always wanted to know about copula modeling but were afraid to ask. *Journal of Hydrologic Engineering*, 12(4):347–368, 2007.

[28] C. Genest, K. Ghoudi, and L. P. Rivest. A semiparametric estimation procedure of dependence parameters in multivariate families of distributions. *Biometrika*, 82(3):543–552, 1995.

[29] Yasser Gonzalez-Fernandez and Marta Soto. copulaedas: An R package for estimation of distribution algorithms based on copulas. *Journal of Statistical Software*, 58(9):1–34, 2014. http://www.jstatsoft.org/v58/i09/.

[30] Yasser Gonzalez-Fernandez and Marta Soto. *copulaedas: Estimation of Distribution Algorithms Based on Copulas*, 2015. R package version 1.4.2, https://CRAN.R-project.org/package=copulaedas.

[31] Yasser Gonzalez-Fernandez and Marta Soto. *vines: Multivariate Dependence Modeling with Vines*, 2016. R package version 1.1.5, https://CRAN.R-project.org/package=vines.

[32] Ronald Greeley, Ruslan O. Kuzmin, and Robert M. Haberle. Aeolian processes and their effects on understanding the chronology of Mars. *Space Science Reviews*, 96(1):393–404, 2001.

[33] T. J. Hastie, R. J. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, New York, 2009.

[34] Rosalyn K. Hayward, Kevin F. Mullins, Lori K. Fenton, Trent M. Hare, Timothy N. Titus, Mary C. Bourke, Anthony Colaprete, and Philip R. Christensen. Mars global digital dune database and initial science results. *Journal of Geophysical Research: Planets*, 112(E11), 2007.

[35] Jerry L. Hintze and Ray D. Nelson. Violin plots: a box plot-density trace synergism. *The American Statistician*, 52(2):181–184, 1998.

[36] Chris H. Hugenholtz and Tom E. Barchyn. Spatial analysis of sand dunes with a new global topographic dataset: new approaches and opportunities. *Earth surface processes and landforms*, 35(8):986–992, 2010.

[37] H. Joe. Families of m-variate distributions with given margins and $m(m-1)/2$ bivariate dependence parameters. In L. Rüschendorf, B. Schweizer, and M. D. Taylor, editors, *Distributions with Fixed Marginals and Related Topics*, pages 120–141, 1996.

[38] H. Joe. *Multivariate Models and Dependence Concepts*. Chapman & Hall, 1997.

[39] E. J. Keogh and M. J. Pazzani. Learning augmented bayesian classifiers: A comparison of distribution-based and classification-based approaches, 1999.

[40] I. Kojadinovic and J. Yan. Comparison of three semiparametric methods for estimating dependence parameters in copula models. *Insurance: Mathematics and Economics*, 47(1):52–63, 2010.

[41] D. Kurowicka and R. M. Cooke. *Uncertainty Analysis with High Dimensional Dependence Modelling*. John Wiley & Sons, 2006.

[42] R. A. Loaiza-Maya, Gomez-Gonzalez J. E., and L. F. Melo-Velandia. Latin american exchange rate dependencies: A regular vine copula approach. *Contemporary Economic Policy*, 33(3):535–549, 2015.

[43] E. D. McKee. Introduction to a study of global sand seas. In McKee E. D., editor, *Copulae in mathematical and quantitative finance*, pages 1–19. University Press of the Pacific, 1979.

[44] D. Muller and C. Czado. Representing sparse gaussian dags as sparse rvines allowing for non-gaussian dependence. *Journal of Computational and Graphical Statistics*, 2017.

[45] T. Nagler, C. Bumann, and C. Czado. Model selection in sparse high dimensional vine copula models with application to portfolio risk. *arXiv:1801.09739l*, 2018.

[46] R. B. Nelsen. *An Introduction to Copulas*. Springer, 2 edition, 2006.

[47] R. S. Olson, W. La Cava, Z. Mustahsan, A. Varik, and J.H . Moore. Data-driven advice for applying machine learning to bioinformatics problems. *arXiv:1708.05070*, 2017.

[48] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825–2830, 2011.

[49] D. M. W. Powers. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. *Journal of Machine Learning Technologies*, 2(1):37–63, 2011.

[50] B. Ripley. *Pattern recognition and neural networks*. Cambridge University Press, 1996.

[51] J. D. Rodriguez, A. Perez, and J. A. Lozano. Sensitivity analysis of k-fold cross validation in prediction error estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(3):569–575, 2010.

[52] U. Schepsmeier. Maximum likelihood estimation of c-vine pair-copula constructions based on bivariate copulas from different families, 2010.

[53] U. Schepsmeier, J. Stoeber, E. C. Brechmann, and B. Graeler. *VineCopula: statistical inference of vine copulas*, 2018.

[54] Gideon Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 1978.

[55] S. Silvestro, L. K. Fenton, D. A. Vaz, N. T. Bridges, and G. G. Ori. Ripple migration and dune activity on Mars: Evidence for dynamic wind processes. *Geophysical Research Letters*, 37(20), 2010.

[56] A. Sklar. Fonctions de repartition à n dimensions et leurs marges. *Publications de l'Institut de Statistique de l'Universite de Paris*, 8:229–231, 1959.

[57] A. Sklar. Random variables, joint distribution functions, and copulas. *Kybernetica*, 9:449–460, 1973.

[58] Marta Soto, Alberto Ochoa, Yasser Gonzalez-Fernandez, Yanely Milanes, Adriel Alvarez, Diana Carrera, and Ernesto Moreno. Vine estimation of distribution algorithms with application to molecular docking. In S. Shakya and R. Santana, editors, *Markov Networks in Evolutionary Computation*, volume 14 of *Adaptation, Learning, and Optimization*, pages 209–225. Springer, 2012. ISBN 978-3-642-28899-9.

[59] Jakob Stoeber, Harry Joe, and Claudia Czado. Simplified pair copula constructions - limitations and extensions. *Journal of Multivariate Analysis*, 119:101–118, 2013.

[60] Lavanya Sita Tekumalla, Vaibhav Rajan, and Chiranjib Bhattacharyya. Vine copulas for mixed data: multi-view clustering for mixed data beyond meta-gaussian dependencies. *Machine Learning*, pages 1–27, 2017.

[61] V. N. Vapnik. *The nature of statistical learning theory*. Springer, 2000.

[62] David A Vaz, Pedro T. K. Sarmento, Maria T. Barata, Lori K. Fenton, and Timothy I. Michaels. Object-based dune analysis: Automated dune mapping and pattern characterization for Ganges Chasma and Gale crater, Mars. *Geomorphology*, 250:128–139, 2015.

[63] Sharon A. Wilson and James R. Zimbelman. Latitude-dependent nature and physical characteristics of transverse aeolian ridges on Mars. *Journal of Geophysical Research: Planets*, 109(E10), 2004.

[64] Hsiang-Fu Yu, Fang-Lan Huang, and Chih-Jen Lin. Dual coordinate descent methods for logistic regression and maximum entropy models. *Machine Learning*, 85(1):41–75, 2011.

[65] Zhi-Hua Zhou. *Ensemble methods: foundations and algorithms*. CRC Press, 2012.