unibz

Fakultät für Wirtschaftswissenschaften
Facoltà di Economia
Faculty of Economics and Management

BEMPS –
Bozen Economics & Management
Paper Series

NO 32/ 2015

Clustering dependent observations with copula functions

F. Marta L. Di Lascio, Simone Giannerini

Clustering dependent observations with copula functions

F. Marta L. Di Lascio[*] Simone Giannerini[†]

Abstract

This paper deals with the problem of clustering dependent observations according to their underlying complex generating process. Di Lascio and Giannerini (2012) introduced the CoClust, a clustering algorithm based on copula function that achieves the task but has a high computational burden. Moreover, the CoClust automatically allocates all the observations to the clusters; thus it cannot discard potentially irrelevant observations. In this paper we introduce an improved version of the CoClust that both overcomes these issues and performs better in many respects. By means of a Monte Carlo study we investigate the features of the algorithm we propose and show that it improves consistently with respect to the old CoClust. The validity of our proposal is also supported by applications to real data sets of human breast tumor samples for which the algorithm provides a meaningful biological interpretation. The new algorithm is implemented and made available through an updated version of the `R` package `CoClust`.

Keywords: Copula function, Multivariate dependence structure, Clustering, Biological tumor sample.
JEL codes: C10, C38.

[*] Email: marta.dilascio@unibz.it. Faculty of Economics and Management, University of Bozen-Bolzano, Italy.

[†] Email: simone.giannerini@unibo.it. Department of Statistical Sciences, University of Bologna, Italy.

1 Introduction

Clustering is a useful exploratory technique as it groups similar objects, e.g. observations or variables, together and makes it possible to identify potentially meaningful relationships between them.

There is an extensive literature on clustering techniques in many different fields of application. In the context of gene expression analysis, a clustering algorithm creates groups of genes, which are involved in the same or in similar biological processes, or groups of biological samples of the same kind. For example, Eisen et al (1998) were among the first to prove the usefulness of hierarchical clustering to reveal biologically meaningful patterns in microarray data; also,Tamayo et al (1999) used self-organizing maps to distinguish two different types of acute leukemia. Nevertheless, it is generally recognized that such methods lack statistical rigour and some crucial questions cannot be addressed, like the number of clusters and the kind of dissimilarity/distance measure to be used in order to uncover differentially expressed genes (Roverato and Di Lascio, 2011; Dortet-Bernadet and Wicker, 2008). The model-based clustering (MClust hereafter) approach (Fraley and Raftery, 1998) is an alternative to distance-based algorithms. It assumes that the data are generated by a mixture of probability distributions such as multivariate normal. Yeung et al (2001) showed among the first that, in general, the use of Gaussian mixture models performs well for clustering gene expressions. However, the MClust only accounts for a linear dependence relationship between objects so that it inherits all the limitations of the linear correlation coefficient as a dependence measure. First, zero correlation does not imply independence. Given two random variables, say X and Y, zero correlation only requires null covariance $cov[X, Y] = 0$, whereas zero dependence requires $cov[\phi_1(X), \phi_2(Y)] = 0$ for any functions ϕ_1 and ϕ_2. Second, linear correlation is not defined for some heavy-tailed distributions whose second moments do not exist, e.g., Student's t distribution with 1 or 2 degrees of freedom. Third, it is not invariant under strictly increasing nonlinear transformations, and, fourth, attainable values of the correlation coefficient within the interval $[-1, +1]$ depend upon their respective marginal distributions F_1 and F_2. Finally, the linear correlation coefficient, being a pairwise measure, cannot account for possible multivariate dependencies. These limitations motivate the introduction of more meaningful dependence measures. The copula function (Sklar, 1959) is a well-known multivariate tool for generating joint distributions with a variety of dependence structures. Hence, starting from the work of Di Lascio and Giannerini (2012), we propose a clustering algorithm based on copula function which inherits some of the advantages of a model-based approach and tries to overcome their disadvantages. The method we discuss here is implemented in the updated version of `R` package `CoClust` available on CRAN (Di Lascio and Giannerini, 2014).

The paper is organized as follows. In Section 2 we introduce the notation used and the theoretical background. In Section 3, we present the copula-based clustering algorithm and describe it in detail. The main novel features of the CoClust are investigated in Section 4 by means of a simulation study. In Section 5, we present an application on a real data set of biological samples of human breast cancer (Hedenfalk et al, 2001). Finally, in Section 6 conclusions and discussion are outlined.

2 Theoretical background

Copula function $C(\cdot)$ is a mathematical object defined in Sklar's theorem (Sklar, 1959). It has a nice probabilistic interpretation since it expresses any K-dimensional joint distribution function $F(\cdot)$ as a combination of standard uniform margins $F_1, \dots, F_k, \dots, F_K$ and the multivariate dependence structure separately. Hence, any joint probability function can be split into the margins and a copula, so that the latter only represents the 'association' between variables. For continuous random variables, the copula density $c(\cdot)$ is related to the density $f(\cdot)$ of the distribution $F(\cdot)$, through the well-known canonical representation $f(x_1, \dots, x_K) = c(F_1(x_1), \dots, F_K(x_K)) \prod_{k=1}^{K} f_k(x_k)$. Hence, the log–likelihood function of $f(\cdot)$ is composed of two positive terms as follows

$$l(\theta) = \sum_{i=1}^{n} \log c \left\{ F_1\left(X_{1i}\right), \dots, F_K\left(X_{Ki}\right); \theta \right\} + \sum_{i=1}^{n} \sum_{k=1}^{K} \log f_i\left(X_{ki}\right)$$

so that estimation can be performed in two steps: *i*) identification of the marginal distributions and *ii*) selection of the appropriate copula function. Clearly, it is possible to model the margins separately from the dependence structure and to use any combination of estimation methods for univariate distributions and copula models. Here we focus on the semi-parametric version of the two–stage estimation method called *inference for the margins* (Joe and Xu, 1996) where the empirical cumulative distribution functions are

Table 1: Definition of some classic single parameter copula functions with corresponding range of the dependence parameter θ and its relation with Kendall's τ. $D_1(x)$ denotes the "Debye" function $1/x \int_0^x t/(\exp^t - 1)dt$.

Copula	$C(u, v; \theta)$	Parameter range	Kendall's τ
Clayton	$\left[\sum_{k=1}^{K} u_k^{-\theta} - K + 1\right]^{-\frac{1}{\theta_2}}$	$\theta \in (0, \infty)$	$\frac{\theta}{\theta+2}$
Frank	$-\frac{1}{\theta_2} \ln \left\{1 + \frac{\prod_{k=1}^{K}(e^{-\theta_2 u_k} - 1)}{(e^{-\theta_2} - 1)^{K-1}}\right\}$	$\theta \in (0, \infty)$	$1 - \frac{4}{\theta}[1 - D_1(\theta)]$
Gaussian	$\Phi_{\mathrm{G}}[\Phi^{-1}(u_1), \Phi^{-1}(u_K)]$	$\theta \in [-1, 1]$	$\frac{2}{\pi} \arcsin(\theta)$
Gumbel	$e^{\left[-\left(\sum_{j=1}^{p} - \log u_j\right)^{1/\theta}\right]}$	$\theta \in [1, \infty)$	$1 - \frac{1}{\theta}$

used to model the margins without assumptions on their parametric form, i.e. through the empirical cumulative distribution function $\hat{F}_k\left(X_{ki}\right)$ with $k = 1, \ldots, K$, and, the maximum likelihood is used to estimate the copula parameter.

In the literature, many different bivariate copula models are available (Nelsen, 2006; Trivedi and Zimmer, 2005) but in higher dimension the Elliptical and the Archimedean families are mostly used. The families considered here are defined in Table 1 and displayed in Figure 1. These make it possible to cover a large set of multivariate features that include asymmetries and heavy tails.

The CoClust algorithm introduced in Di Lascio and Giannerini (2012) creates clusters such that objects in a same cluster are independent, i.e. realizations of the same univariate random variable, while objects belonging to different clusters are dependent, i.e. realizations of a copula model taken as the data generating process (DGP hereafter). Here, a clustering is represented by a multivariate probability model defined via copula whose dimension K is the number of clusters; each cluster represents a (marginal) univariate distribution function $F_k(X_k)$ with $k = 1, 2, \ldots, K$. The approach can bee seen as an MClust in which the DGP is defined via copula and the focus is on the inter-cluster dependence relationship. The main difference with traditional MClust is that here we assume internal independence and external dependence rather than internal similarity and external separation (see Di Lascio and Giannerini (2012) for more details).

The connection between the copula-based clustering as a statistical object and the clustering as a biological object is formalized as follows. The starting

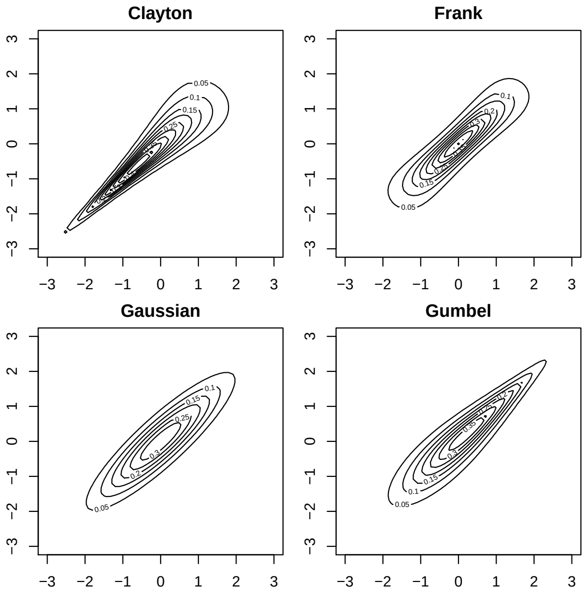

Figure 1: Contour plots of the four (bivariate) copula functions defined in Table 1 with standard normal margins and a Kendall's τ coefficient of 0.7.

point is a $G \times S$ data matrix

$$\begin{bmatrix} x_{11} & \dots & x_{1s} & \dots & x_{1S} \\ \vdots & \dots & \vdots & \dots & \vdots \\ x_{g1} & \dots & x_{gs} & \dots & x_{gS} \\ \vdots & \dots & \vdots & \dots & \vdots \\ x_{G1} & \dots & x_{Gs} & \dots & x_{GS} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_g \\ \vdots \\ \mathbf{x}_G \end{bmatrix} \tag{1}$$

where $\mathbf{x}_g$, $g = 1, \dots, G$ is a row vector containing the expression level of the gene g observed in S biological samples and is a single element to be allocated to a cluster.

The CoClust algorithm allocates a k-plet of row vectors at a time and the allocation of the i-th k-plet of genes, $\mathbf{x}_{g_{1i}}, \dots, \mathbf{x}_{g_{ki}}$, is performed on the basis of the log-likelihood of the copula fit:

$$l_{\substack{\mathbf{x}_{g^*_{11}} \; \dots \; \mathbf{x}_{g^*_{k1}} \\ \dots \quad \dots \; \dots \\ \mathbf{x}_{g^*_{1(i-1)}} \dots \mathbf{x}_{g^*_{k(i-1)}} \\ \mathbf{x}_{g_{1i}} \; \dots \; \mathbf{x}_{g_{ki}}}} \left(\hat{\theta}\right) = \max_{\theta \in \Theta} \sum_{s=1}^{S} \left\{ \sum_{j=1}^{i-1} \log c \left[\hat{F}_{g^*_{1j}} \left(X_{g^*_{1j}s} \right), \dots, \hat{F}_{g^*_{kj}} \left(X_{g^*_{kj}s} \right); \theta \right] + \right.$$
$$\left. \log c \left[\hat{F}_{g_{1i}} \left(X_{g_{1i}s} \right), \dots, \hat{F}_{g_{ki}} \left(X_{g_{ki}s} \right); \theta \right] \right\} \qquad (2)$$

where the asterisk in the subscripts indicates the genes already allocated to the clusters.

3 CoClust: copula–based clustering algorithm

The CoClust assumes that the data are generated by a K-dimensional copula function whose arguments are the probability-integral transforms of the density functions that generate the clusters. At the first step the algorithm selects the number of clusters K; then, it evaluates the allocation of one K-plet of observations at a time, i.e. one observation for each cluster. Recall that each observation is a S–dimensional vector and its components are treated as (independent) realizations of the same random variable. The K candidate observations are allocated to the K clusters on the basis of the value of the maximized log–likelihood function of the copula model. Since at each step we compare non–nested models, the criterion is equivalent to the well–known Bayes information criterion (BIC) and Akaike information criterion (AIC).

The CoClust produces a set of groups with a precise dependence structure: observations that belong to different clusters are dependent through the copula model. From the graph theory point of view, the final clustering of the CoClust can be seen as a complete graph where each vertex indicates a cluster of objects independent from each other, each edge indicates the dependence relationship between each pair of clusters and the undirected distinct vertices indicate the exchangeability of the dependence relationship.

As in Di Lascio and Giannerini (2012), there is no need to set a priori the exact number of clusters K, nor is a starting classification required because the algorithm automatically selects the best number of clusters within a given range of possibilities on the basis of the log-likelihood in eq. (2).

One of the most crucial improvements of our proposal concerns the selection of K in $\{2, 3, \ldots, K_{\max}\}$ where $K_{\max}$ is specified by the user. The algorithm in Di Lascio and Giannerini (2012) chooses the number of clusters K by estimating $\sum_{k=2}^{K_{\max}} C_{G,k} = \binom{G}{k}$ copula fits. In our proposal only $\sum_{k=2}^{K_{\max}} C_{n_k,k} = \binom{n_k}{k}$ fits are required, where $n_k \ll G$. In practice we perform the selection of the number of clusters K on a representative subset of n_k observations and this reduces greatly the computational burden. For more details on the selection of the n_k observations see Section 4.

At the generic i-th step, the algorithm in Di Lascio and Giannerini (2012) selects the candidate k-plet that corresponds to the largest log-likelihood of the copula among those computed on the set of $\binom{G-(i-1)k}{k}$ combinations of observations that have not been allocated yet. In the version we propose the candidate k-plet is constructed on the basis of a function $H(\cdot)$ applied to the row vectors of eq. (1) and defined as follows:

Definition 1 *Let $\Lambda = \{g_1, \ldots, g_h\}$ a set of genes such that $\Lambda = \Lambda_1 \cup \Lambda_2$, where Λ_1 is the subset of genes already selected to compose a k-plet and Λ_2 is the set of remaining candidates to complete it. The function $H(\cdot)$ is defined as follows*

$$H(g_1, \ldots, g_h) = \max_{g' \in \Lambda_2} \left\{ \underset{g \in \Lambda_1}{\psi} \left(\mathrm{cor}(\mathbf{x}_g, \mathbf{x}_{g'}) \right) \right\} \tag{3}$$

where $\mathbf{x}_g$ is the expression level of the gene g, cor *is the Spearman's correlation coefficient and ψ is a convenient function among the mean, the median or the maximum.*

The function $H(\cdot)$ is a sort of multivariate measure of dependence based on the pairwise Spearman's correlations. When the clustering concerns biological samples, then the function $H(\cdot)$ is applied to the column vectors of the data matrix in eq. (1).

Differently from Di Lascio and Giannerini (2012), it is now possible to discard irrelevant objects since the permutation of the selected k-plet is allocated if and only if it increases the likelihood of the copula fit in eq. (2); otherwise, it is discarded since, possibly, either it is independent from the identified DGP or it comes from another DGP.

In the following we describe the procedure of the improved CoClust. The data are organized in a $G \times S$ matrix like in eq. (1). Let C be a copula model and $l_C(\cdot)$ the associated log-likelihood function and, further, define as $\Psi[A]$

Algorithm 1 Copula-based clustering algorithm (Part 1)

Input: $G \times S$ data matrix, copula model $C(\cdot)$, estimation method for copula and margins, a selection criterion (e.g. BIC), $[K_{\min}, K_{\max}]$ range for the number of clusters, dimension n_k of the initial subset of rows used to select K, the function ψ in eq. (3) (max, mean or median).

Output: Matrix of the clustered data with K columns, estimated copula model, log-likelihood of copula fit.

```
Compute the Spearman's correlation matrix between observations $\mathbf{x}_g, \mathbf{x}_{g'}$: $\text{cor}(g, g')$ with $g \neq g'$, and $g, g' \in \{1, \dots, G\}$
for $k = K_{\min}, \dots, K_{\max}$ do
    set $i = 1$
    while $(i \leq n_k)$ do
        if $i = 1$ then
            $\text{allocated}_{i-1} = \emptyset$
        else
            $\text{allocated}_{i-1} = \{1, \dots, G\} \backslash \{(g_{11}, \dots, g_{k1}), \dots, (g_{1(i-1)}, \dots, g_{k(i-1)})\}$;
            select $(g_{1i}, g_{2i}) = \arg\max_{\{1,\dots,G\} \backslash \text{allocated}_{i-1}} \text{cor}(g, g')$;
            set $j = 2$;
        end if
        while $(j < k)$ do
            set $\Lambda = (\{g_{1i}, \dots, g_{ji}\}) \cup (\{1, \dots, G\} \setminus \text{allocated}_{i-1} \setminus \{g_{1i}, \dots, g_{ji}\})$;
            compute $H(\Lambda)$ and obtain $g_{(j+1)i}$;
            $j = j + 1$;
        end while
        set $\text{candidate}_i = (g_{1i}, \dots, g_{ki})$
        if $i = 1$ then
            $\text{allocated}_i = \text{candidate}_i$;
            $i = i + 1$;
        else
            if $\max_{\Psi[\text{candidate}_i]} l_C(\text{allocated}_{i-1} \cup \text{candidate}_i) \geq l_C(\text{allocated}_{i-1})$ then
                allocated$= \arg\max_{\Psi[\text{candidate}]} l_C(\text{allocated}_{i-1} \cup \text{candidate}_i)$;
                $\text{allocated}_i = \text{allocated}_{i-1} \cup$ allocated
                $i = i + 1$;
```

Algorithm 1 Copula-based clustering algorithm (Part 2)

```
26:             else
27:                 reject the candidate
28:             end if
29:         end if
30:     end while
31:     Compute the selection criterion $\mathrm{BIC}_k$ for $C$ on the $n_k$ allocated $k$-plets
32: end for
33: $K = \arg\max_k \mathrm{BIC}_k$
34: set $k = K$
35: for $i = (n_K + 1), \dots, G/K$ do
36:     repeat steps 5–29
37: end for
```

the set of permutations of the elements of the set A. The detailed procedure of the algorithm is shown in the box Algorithm 1.

Note that the choice of the copula model is left to the user as the modelling step always requires a careful assessment. We argue that for a clustering exercise to be robust, the results should be insensitive to the choice of similar models, provided they are appropriate for the case under scrutiny. Copula models with a single dependence parameter can be thought of as non-nested models (except the case of Gaussian copula with unstructured covariance matrix). Hence, as suggested by Zimmer and Trivedi (2006), one approach for choosing among copula models is to use either the AIC or BIC. In our case, AIC$= -2l(\hat{\theta}) + 2$ is equivalent to the maximized log-likelihood of the copula since the number of parameters is always 1 (single-parameter copulas and nonparametric estimation of the margins). On the contrary, BIC$= -2l(\hat{\theta}) + log(n)$, where n is the total number of allocated observations, is useful when the selected number of clusters varies among copula models.

3.1 A toy example

Here we present an example of the CoClust algorithm that should help clarifying its logic. Suppose we have a $G \times S$ data matrix as in eq. (1) where $G = 9$. Given a copula model $C(\cdot)$, a given number of clusters $K = 3$ and $n_K = 2$, we

1. compute the Spearman's correlation matrix;

2. since $i = 1 \leq n_K = 2$, the first pair of rows is selected among the $\binom{9}{2} = 36$ available: $(g_{11}, g_{21}) = \arg\max_{\{1,2,\dots,9\}} \mathrm{cor}(g, g')$, say $(\mathbf{x}_1, \mathbf{x}_3)$;

3. since $j = 2 < K = 3$, the triplet is completed by selecting the third row on the basis of the $H(\cdot)$ function in eq. (3) where $\Lambda = \{1, \dots, 9\}$, $\Lambda_1 = \{1, 3\}$ and $\Lambda_2 = \{2, 4, \dots, 9\}$:

$$H\left(\{1,3\} \cup \{2,4,5,6,7,8,9\}\right) = \max \begin{bmatrix} \psi(\mathrm{cor}(\mathbf{x}_1, \mathbf{x}_2)) \\ \psi(\mathrm{cor}(\mathbf{x}_1, \mathbf{x}_4)) \\ \vdots \\ \psi(\mathrm{cor}(\mathbf{x}_1, \mathbf{x}_9)) \\ \psi(\mathrm{cor}(\mathbf{x}_3, \mathbf{x}_2)) \\ \vdots \\ \psi(\mathrm{cor}(\mathbf{x}_3, \mathbf{x}_9)) \end{bmatrix}$$

   and $(g_{11}, g_{21}, g_{31}) = \arg H(1, \dots, G)$; say that the first selected triplet is $(\mathbf{x}_1, \mathbf{x}_3, \mathbf{x}_4)$;

4. $j = j + 1 = 3 \nless K = 3$, hence the triplet candidate is $(\mathbf{x}_1, \mathbf{x}_3, \mathbf{x}_4)$; since $i = 1$, the candidate is allocated to the 3 clusters obtaining the following initial classification: $\mathrm{C}_1 = \mathbf{x}_1$, $\mathrm{C}_2 = \mathbf{x}_3$, $\mathrm{C}_3 = \mathbf{x}_4$; next, i is updated to $i + 1 = 2$;

5. the procedure is repeated from step 5. to step 21. one time: $i = 2 \leq n_K = 2$, a second doublet is selected: $(g_{12}, g_{22}) = \arg\max_{\{2,5,6,\dots,9\}} \mathrm{cor}(g, g')$, say $(\mathbf{x}_2, \mathbf{x}_6)$;

6. $j = 2 < K = 3$, the triplet is completed by selecting the third row on the basis of the $H(\cdot)$ function in eq. (3), that is

$$H(\{2,6\} \cup \{4,5,7,8,9\}) = \max \begin{bmatrix} \psi(\mathrm{cor}(\mathbf{x}_2, \mathbf{x}_4)) \\ \vdots \\ \psi(\mathrm{cor}(\mathbf{x}_2, \mathbf{x}_9)) \\ \psi(\mathrm{cor}(\mathbf{x}_6, \mathbf{x}_4)) \\ \vdots \\ \psi(\mathrm{cor}(\mathbf{x}_6, \mathbf{x}_9)) \end{bmatrix}$$

Table 2: Final clustering of the toy example.

C_1	C_2	C_3
$\mathbf{x}_1$	$\mathbf{x}_3$	$\mathbf{x}_4$
$\mathbf{x}_6$	$\mathbf{x}_2$	$\mathbf{x}_5$

and $(g_{12}, g_{22}, g_{32}) = \arg H(\{2,4,5,6,7,8,9\})$; say that the second selected triplet is $(\mathbf{x}_2, \mathbf{x}_6, \mathbf{x}_5)$ and its is the candidate to the allocation since $j = j + 1 = 3 \nless K = 3$;

7. in order to choose whether either to allocate or to discard the second triplet, the copula model is estimated through eq. (2) and according to steps 19.-21. In practice, the following log-likelihood is computed

$$l_{\substack{\mathbf{x}_{g^*_{11}},\mathbf{x}_{g^*_{21}},\mathbf{x}_{g^*_{31}} \\ \mathbf{x}_{g_{12}},\mathbf{x}_{g_{22}},\mathbf{x}_{g_{32}}}}\left(\hat{\theta}\right) = l_{\substack{\mathbf{x}_1,\mathbf{x}_3,\mathbf{x}_4 \\ \mathbf{x}_2,\mathbf{x}_6,\mathbf{x}_5}}\left(\hat{\theta}\right) \tag{4}$$

   by varying the permutation of the second triplet candidate to the allocation; if the maximum value of the maximized loglikelihood is greater than $l_{\mathbf{x}_{g^*_{11}},\mathbf{x}_{g^*_{21}},\mathbf{x}_{g^*_{31}}}\left(\hat{\theta}\right)$ then the second triplet is allocated; say that $l_C((\mathbf{x}_1, \mathbf{x}_6), (\mathbf{x}_3, \mathbf{x}_2), (\mathbf{x}_4, \mathbf{x}_5)) \geq l_C(\mathbf{x}_1, \mathbf{x}_3, \mathbf{x}_4)$, the classification obtained is $\mathrm{C}_1 = (\mathbf{x}_1, \mathbf{x}_6)$, $\mathrm{C}_2 = (\mathbf{x}_3, \mathbf{x}_2)$, $\mathrm{C}_3 = (\mathbf{x}_4, \mathbf{x}_5)$ and i is updated to $i = 2$;

8. now, steps 5.-21. are repeated; say that the third candidate triplet is $(\mathbf{x}_7, \mathbf{x}_8, \mathbf{x}_9)$ and that

$$\max_{\Psi[(\mathbf{x}_7,\mathbf{x}_8,\mathbf{x}_9)]} l_C((\mathbf{x}_1, \mathbf{x}_3, \mathbf{x}_4) \cup (\mathbf{x}_7, \mathbf{x}_8, \mathbf{x}_9)) \ngeq l_C(\mathbf{x}_1, \mathbf{x}_3, \mathbf{x}_4)$$

   then the third triplet is rejected. The final clustering is presented in tab. 2 and the dependence relationship across clusters is estimated.

4 Simulation study

We investigate the performance of the CoClust through a large simulation study where we explore the features of the new version of the CoClust and compare it with that in Di Lascio and Giannerini (2012). The first part of the simulation study focuses on the observations and k-plets correctly allocated. In the second part we vary

- the kind of function H: max, mean, and median;
- the number of clusters/dimension of the copula: from 2 to 5;
- the copula model: Archimedean family, Gaussian family.

Moreover, we study the capability of the CoClust to distinguish between different DGPs in the same data set and possibly dropping out irrelevant observations. At this stage, we chose not to vary the sample size mainly because the CoClust is designed to work with small samples. Also, high-dimensional copula models, e.g. $K > 10$, are not easily tractable from the computational point of view. Hence, in all the simulated examples the sample size is set to $n = 60$. The number of Monte Carlo replications is 500 in all the scenarios.

4.1 Performance measures

In order to evaluate the ability of an algorithm to detect objects that belong to an underlying DGP or to exclude objects not belonging to it we use measures like sensitivity and positive predictive value, commonly employed for diagnostic tests. Here, the sensitivity is the proportion of objects belonging to the DGP that will be clustered; the positive predictive value is the proportion of clustered objects that actually belong to the true DGP. From Table 3 we can compute the sensitivity with $n_{11}/n_{1\cdot}$ and the positive predictive value with $n_{11}/n_{\cdot 1}$, since $n_{i\cdot}$ with $i = 1, 2$ indicates the number of k-plets (or observations) to be allocated or discarded, while $n_{\cdot j}$ with $j = 1, 2$ indicates the number of k-plets (or observations) that the CoClust has correctly or incorrectly allocated or discarded. In summary, we use the following performance measures:

Table 3: Confusion matrix of the clustering algorithm.

	Allocated k-plets (or obs.)	Discarded k-plets (or obs.)	
true DGP	n_{11}	n_{12}	$n_{1\cdot}$
other DGP	n_{21}	n_{22}	$n_{2\cdot}$
	$n_{\cdot 1}$	$n_{\cdot 2}$	n

1. *p.n.c.*: percentage of replications in which the identified number of clusters is correct;
2. *SEN.k*: sensitivity for k-plets; percentage of k-plets of objects belonging to the true DGP that are correctly allocated;
3. *SEN.o*: sensitivity for observations; percentage of single objects belonging to the true DGP that are correctly allocated;
4. *PPV.k*: positive predictive value of k-plets; percentage of clustered k-plets of objects that belong to the true DGP;
5. *PPV.o*: positive predictive value of observations; percentage of clustered objects that belong to the true DGP.

4.2 Comparing the two versions of the CoClust algorithm

The design of the simulation study follows that of Section 4 of Di Lascio and Giannerini (2012) as to facilitate the comparison. Recall that here $n = 30$ and $K = 3$ for all the scenarios. Table 4 shows the results obtained by using our proposal. The copula generating process is reported in the first column and the margins are reported in the second column. Note that *p.n.c.* and *SEN.o* coincide, respectively, with the measures *p.n.c.* and *p.c.a* in Di Lascio and Giannerini (2012). By comparing the results in Table 4 with those

Table 4: CoClust performance: comparing our proposal with the CoClust in Di Lascio and Giannerini (2012). The symbol $\star$ indicates where the new algorithm improves with respect to the old CoClust in Di Lascio and Giannerini (2012).

Copula	Margins	*p.n.c.*	*SEN.k*	*SEN.o*	*PPV.k*	*PPV.o*
Frank copula	Gamma, Beta, Gaussian	$100^\star$	99.50	$99.83^\star$	99.50	99.83
Skew–Normal	Skew–Normal	$92.50^\star$	82.03	84.46	95.24	98.45
Mixed Gaussian	Gaussian	$71.25^\star$	79.00	$78.30^\star$	84.25	94.47

in tables 6, 7, 8 in Di Lascio and Giannerini (2012) we can see that the new algorithm overcomes the previous version. The performance gain is considerable when the data come from a mixture of Gaussian distributions. In general, the obtained results for all the performance measures employed are very satisfactory in all the three scenarios.

4.3 Understanding the ability of the CoClust

In this section we investigate the features of the new CoClust algorithm. In the first scenario we simulate from a Skew–Normal DGP by varying the number of clusters K in $(2, 3, 4, 5)$ and the function ψ in eq. (3). We use a Gumbel copula when $K = 2$ and a Clayton copula when $K = 3, 4, 5$, the correlation between any pair of margins is set to 0.7, the mean vector is set to $(4, 6)$, $(4, 6, 7)$, $(2, 4, 6, 7)$ and $(3, 5, 7, 9, 11)$, respectively for each value of K from 2 to 5. Table 5 shows the results. As for the correct number of clusters (*p.n.c.*), the performance of the CoClust is satisfying since all the

Table 5: CoClust performance: Skew-Normal DGP for $K = 2, \dots, 5$ and the function ψ in eq. (3).

K	ψ	*p.n.c.*	*SEN.k*	*SEN.o*	*PPV.k*	*PPV.o*
2	max	80.80	81.53	83.82	97.17	99.99
	mean	86.60	81.62	83.90	97.19	99.99
	median	92.00	81.83	84.13	97.16	99.99
3	max	82.40	69.77	76.00	89.28	97.40
	mean	80.20	73.60	79.31	90.63	97.77
	median	84.80	73.76	79.26	90.96	97.87
4	max	95.80	84.48	88.17	93.90	98.22
	mean	96.80	89.02	91.91	95.51	98.78
	median	96.40	87.00	90.32	94.86	98.67
5	max	84.40	71.33	80.26	82.15	92.54
	mean	85.80	78.92	85.45	86.38	93.89
	median	84.00	77.24	84.44	85.50	93.78

percentages are well above 80%; note that the function ψ has some impact on the identification of the true value of K: when the number of clusters is low, i.e. $K \leq 3$, the median outperforms the maximum and the mean, while when $K \geq 4$, the mean function is slightly better. On the one side, the algorithm appears to be quite sensitive: in the worst case it discards (or incorrectly allocates) less than three k-plets over ten. On the other side, the accuracy of the algorithm is very high: in almost all the scenarios it is greater than 90%. This means that the improved CoClust correctly composes and allocates almost all the k-plets (objects) to the final clustering.

Table 6: CoClust performance: trivariate Skew-Normal DGP plus 15% of independent observations.

Copula	ψ	*p.n.c.*	*SEN.k*	*SEN.o*	*PPV.k*	*PPV.o*
Gaussian	max	83.00	68.39	72.09	76.71	81.12
	mean	83.00	70.93	74.59	77.62	81.86
	median	85.40	71.26	74.85	77.78	81.91
Frank	max	83.80	68.26	73.05	73.70	79.09
	mean	83.40	70.76	75.47	74.60	79.78
	median	86.40	70.91	75.60	74.61	79.73
Gumbel	max	80.80	68.21	72.80	73.78	78.98
	mean	83.80	70.76	75.45	74.63	79.77
	median	85.20	70.93	75.58	74.65	79.73
Clayton	max	78	67.92	72.66	77.47	83.18
	mean	77.2	70.68	75.25	78.36	83.69
	median	81.8	70.69	75.20	78.33	83.58

In Table 6 we show the results for a trivariate Skew-Normal DGP plus 15% of independent observations. The results confirm that the median is the best aggregating function when $K = 3$. As for the sensitivity and the positive predictive value, we may argue that also in this set of scenarios the final clustering is more precise than sensitive: the clustering is 'clean' since few k-plets are incorrectly allocated, but a bit incomplete in that some "good"

k-plets are discarded. Finally, the kind of copula model used in the algorithm does not have a strong impact on the performance of the CoClust.

In Table 7 we show the case of two competing trivariate DGPs, a Skew-Normal DGP with correlation 0.7 and a Clayton copula DGP with $\theta = 2$ (Kendall's correlation is $\tau = 0.5$). Here we assess the capability of the CoClust to recognize and allocate the observations coming from the Skew-normal distribution. Also in this last scenario the CoClust has a high percentage of correctly identified number of clusters (it varies between 85% and 90%), irrespectively of the kind of copula model and the function ψ. Moreover, about 75% of the k-plets and 80% of the observations are correctly allocated to the clusters so that the final clustering is fairly complete. On the contrary, the PPV is not very high as the final clustering contains about 50% of k-plets that not belong to the Skew-normal distribution. This result is somehow expected since both the Clayton and the Skew-normal are heavy tailed on the left. In this scenarios the CoClust is more sensitive than precise

Table 7: CoClust performance: two trivariate data generating processes (Clayton copula and Skew-normal).

Copula	ψ	$p.n.c.$	$SEN.k$	$SEN.o$	$PPV.k$	$PPV.o$
Gaussian	Max	84.4	72.61	77.97	43.68	47.01
	Mean	86.6	74.64	79.68	43.99	47.04
	Median	86.6	75.24	80.18	44.20	47.21
Frank	Max	77.2	72.88	79.64	43.25	47.38
	Mean	80.0	75.28	82.00	43.32	47.28
	Median	84.4	75.21	81.92	43.37	47.32
Gumbel	Max	79.4	71.31	74.85	42.36	44.58
	Mean	82.2	74.16	77.42	42.99	44.97
	Median	84.8	74.41	77.61	43.07	45.01
Clayton	Max	86.2	72.65	81.89	44.08	49.74
	Mean	89.2	74.82	83.44	44.58	49.76
	Median	90.4	74.76	83.43	44.54	49.74

so that the obtained clustering is complete but “dirty”. Finally, in all the performed simulations the number n_k of k-plets used to select the number of clusters K has been set to 4. A small simulation study revealed that the value for n_k does not affect the performance of the algorithm. The `R` code for the simulation experiment and the results that have not been reported here are available upon request.

5 Real data analysis

We apply the CoClust to a data set that contains gene expression levels observed in 21 human breast cancer biological samples with three different kinds of breast cancer mutation: BRCA1, BRCA2, and Sporadic (Hedenfalk et al, 2001). We focus on the 51 genes whose variation in expression among all experiments best differentiated among the three types of tumours. The purpose is the identification of the kind of mutation among the three, so the CoClust is applied to the 21 biological samples. Accordingly to the simulation study, we

- apply the CoClust to the whole dataset (three kinds of mutations);
- apply the CoClust to the three mutations groups separately. Moreover, we add independent random observations and data from different DGPs (mutations) to the data set.

First, we apply the method to the whole data set of biological samples. Table 8 shows the results: the CoClust identifies correctly the number of clusters ($K = 7$) and distinguishes perfectly the three mutations. Also it uncovers the relationship among the samples with the same kind of biological mutations. Of course, we do not know the true composition of the margins and, consequently, we cannot evaluate them but we know the three kinds of mutations and we see that the CoClust makes it possible to recognize them all by looking at the results across clusters. The copula model selected on the basis of the BIC is the Gaussian with $\hat{\theta} = 0.780$ and $\mathrm{SE}\left(\hat{\theta}\right) = 0.028$.

Now we work with a kind of mutation at a time. Table 9 shows that, for each mutation, the CoClust is always able to recognize the correct number of clusters and the true final clustering, also when we add a set of independent observations (Indep. case) or a subset of biological samples with other kinds of cancer mutations (Mix. case), except for the case of Sporadic cancer

Table 8: Identification of breast cancer mutations through CoClust.

C_1	C_2	C_3	C_4	C_5	C_6	C_7
BRCA1	BRCA1	BRCA1	BRCA1	BRCA1	BRCA1	BRCA1
BRCA2	BRCA2	BRCA2	BRCA2	BRCA2	BRCA2	BRCA2
Spor	Spor	Spor	Spor	Spor	Spor	Spor

samples mixed with the other two kinds of mutations. In this case, the zero value for the sensitivity and the positive predictive value is due to the fact that the CoClust never identifies the correct number of clusters (in 88% it selects $K = 9$ and in the remaining 12% of cases it selects $K = 8$). This result is very interesting: the CoClust is able to identify mutations in germ cells, which occur in egg and sperm cells and can be passed on from a parent to a child, rather than mutations occurring in somatic cells, which cannot be inherited. For completeness, the copula model selected for all the three data sets is Gaussian with $\hat{\theta} = 0.797$ and $\text{SE}\left(\hat{\theta}\right) = 0.048$ for the samples with BRCA1 mutation, $\hat{\theta} = 0.850$ and $\text{SE}\left(\hat{\theta}\right) = 0.040$ for the sample with BRCA2 mutation and $\hat{\theta} = 0.574$ and $\text{SE}\left(\hat{\theta}\right) = 0.071$ for the samples with Sporadic mutation.

6 Summary and Discussion

In this paper we have introduced an improved version of the CoClust algorithm in Di Lascio and Giannerini (2012). The main innovative features are (*i*) the capability to discard irrelevant observations, (*ii*) the use of a multivariate function of pairwise correlations to form the k-plets candidate to the allocations, and (*iii*) the improvement in terms of computational complexity. Also for the improved CoClust, the computational complexity is governed by the first step of the algorithm but we pass from a complexity of $\sum_{k=2}^{K_{\max}} \binom{G}{k} \leq (G+1)\left[(G+1)^{k-1} - 1\right] \approx O\left(G^{K_{\max}}\right)$, where G is the sample size, to a complexity of $\sum_{k=2}^{K_{\max}} \binom{n_k}{k} \leq (n_k+1)\left[(n_k+1)^{k-1} - 1\right] \approx O\left(n_k^{K_{\max}}\right)$, where n_k is a user defined constant that does not depend on the sample

Table 9: Analysis of cancer mutations through CoClust.

Copula	ψ	*p.n.c.*	*SEN.k*	*SEN.o*	*PPV.k*	*PPV.o*
BRCA1	Whole	100	100	100	100	100
	Indep.	100	100	100	100	100
	Mix.	100	100	100	100	100
BRCA2	Whole	100	100	100	100	100
	Indep.	100	100	100	100	100
	Mix.	100	100	100	100	100
Sporadic	Whole	100	100	100	100	100
	Indep.	100	100	100	100	100
	Mix.	0	0	0	0	0

size. The superiority of our proposal has been shown both theoretically and empirically. Moreover, the algorithm has been implemented in an R package (Di Lascio and Giannerini, 2014) which is available on CRAN.

The algorithm can be extended in different directions. First of all, the introduction of an automatic selection copula model. As pointed out by Brechmann and Schepsmeier (2013), the tests proposed by Vuong (1989) and Clarke (2007), which are essentially likelihood-ratio-based tests which measure the distance between two statistical models, may be more reliable than information criteria when non-nested models are compared. Second, the limitation of equal sized clusters can be levied by resorting to copula-based imputation methods, e.g. see Di Lascio et al (2015). Third, an investigation of the performance of the CoClust with non exchangeable copulas and rotated copulas could be useful for particular contexts or empirical applications. Another important point concerns the introduction of an ad hoc validation measure for measuring the goodness of a clustering; one could define a copula-based silhouette index or borrow a measure from graph theory.

Acknowledgements

F. Marta L. Di Lascio acknowledges the support of Free University of Bozen-Bolzano, Faculty of Economics and Management, via the project "Multivariate analysis techniques based on copula function".

References

Brechmann E, Schepsmeier U (2013) Modeling dependence with c- and d-vine copulas: The r package cdvine. Journal of Statistical Software 52(3):1–27

Clarke K (2007) A simple distribution-free test for non-nested model selection. Political Analysis 15:347–363

Di Lascio F, Giannerini S (2012) A copula-based algorithm for discovering patterns of dependent observations. Journal of Classification 29(1):50–75

Di Lascio F, Giannerini S (2014) CoClust. R package version 0.3-1

Di Lascio F, Giannerini S, Reale A (2015) Exploring copulas for the imputation of complex dependent data. Statistical Methods & Applications 24(1):159–175

Dortet-Bernadet JL, Wicker N (2008) Model-based clustering on the unit sphere with an illustration using gene expression profiles. Biostatistics 9(1):66–80

Eisen MB, Spellman PT, Brown PO, Botstein D (1998) Cluster analysis and display of genome–wide expression patterns. Proceedings of the National Academy of Sciences 95:14,863–14,868

Fraley C, Raftery A (1998) How many clusters? which clustering method? answers via model–based cluster analysis. The Computer Journal 41(8):578–588

Hedenfalk I, Duggan D, Chen Y, Radmacher M, Bittner M, Simon R, Meltzer P, Gusterson B, Esteller M, kallioniemi OP, Wilfond B, Borg A, Dougherty E, Kononen J, Bubendorf L, Fehrle W, Pittaluga S, Gruvberger S, Loman N, Johannsson O, Olsson H, Sauter G (2001) Gene–expression profiles in hereditary breast cancer. The New England Journal of Medicine 344(8):539–548

Joe H, Xu J (1996) The estimation method of inference functions for margins for multivariate models. Technical Report 166, Department of Statistics, University of British Columbia

Nelsen RB (2006) Introduction to copulas. Springer, New York

Roverato A, Di Lascio F (2011) Wilks' λ dissimilarity measures for gene clustering: An approach based on the identification of transcription modules. Biometrics 67(4)

Sklar A (1959) Fonctions de répartition à n dimensions et leurs marges. Publ Inst Statist Univ Paris 8:229–231

Tamayo P, Slonim D, Mesirov J, Zhu Q, Kitareewan S, Dmitrovsky E, Lander E, Golub T (1999) Interpreting patterns of gene expression with self–organizing maps: methods and application to hematopoietic differentiation. National Academy of Sciences of the United States of America (PNAS) 96:2907–2912

Trivedi PK, Zimmer DM (2005) Copula Modeling: An Introduction for Practitioners, vol 1. Foundations and Trends in Econometrics

Vuong Q (1989) Likelihood ratio tests formodel selection and non-nested hypotheses. Econometrica 57:307–333

Yeung K, Fraley C, Murua A, Raftery A, Ruzzo W (2001) Model-based clustering and data transformations for gene expression data. Bioinformatics 17(10):977–987

Zimmer DM, Trivedi PK (2006) Using trivariate copulas to model sample selection and treatment effects: Application to family health care demand. J Bus Econ Stat 24:63–76