Journal of Computational and Applied Mathematics 461 (2025) 116457

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

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

Outlier detection of multivariate data via the maximization of the cumulant generating function

Francesco Cesarone [a],*, Rosella Giacometti [b], Jacopo Maria Ricci [a]

[a] Roma Tre University - Department of Business Studies, Italy
[b] University of Bergamo - Department of Management, Italy

ARTICLE INFO

Keywords:
Outlier detection
Cumulant generating function
Principal component analysis
Projections
Multivariate skew-normal distribution
Concave programming

ABSTRACT

In this paper, we propose an outlier detection algorithm for multivariate data based on their projections on the directions that maximize the Cumulant Generating Function (CGF). We prove that CGF is a convex function, and we characterize the CGF maximization problem on the unit n-circle as a concave minimization problem. Then, we show that the CGF maximization approach can be interpreted as an extension of the standard principal component technique. Therefore, for validation and testing, we provide a thorough comparison of our methodology with two other projection-based approaches both on artificial and real-world financial data. Finally, we apply our method as an early detector for financial crises.

1. Introduction

Outlier detection issue has become increasingly important over the years and, as of now, its fields of application range from medicine and engineering to finance (see, e.g., [1–3]). As for the latter, outliers can be the consequence of human error or fraudulent activities; similarly, financial crises can be viewed as anomalies since markets experience atypical behaviors in those periods (see, e.g., [4]). Furthermore, a few outliers can strongly influence the results of an experiment. This can be observed in Portfolio Optimization, where some portfolio selection models can be highly sensitive to changes in input data (see, e.g., [5–7]). Because of this widespread practical relevance, many authors tackled this topic. Hence, the theory behind anomaly detection has unsurprisingly evolved, from the first studies which dealt with more simple instances, i.e., univariate Gaussian data, to more complex cases, such as multivariate data following non-parametric distributions. We can mention the work of Ferguson [8], who considers univariate normal samples and identifies outliers as data with mean slippage. Wilks [9] proposes a method to identify outliers in a multivariate normal distribution with unknown parameters. Gnanadesikan and Kettenring [10] propose outlier detection methods on multivariate data, based on their projection onto the directions corresponding to the principal components obtained by the standard Principal Component Analysis (PCA). Schwager and Margolin [11] extend the work of Ferguson [8] to multivariate normal data. Several works address the problem of outlier detection via the "local influence" analysis framework. Starting from the work by Cook [12], which studies the local influence based on the likelihood displacement, Shi and Wang [13] extend Cook's approach to multivariate data by replacing the likelihood displacement with the Mahalanobis distance. Then, Shi [14] generalizes the statistic developed by Cook and Weisberg [15] to study local influence in PCA, showing that his method and that of Cook's are equivalent under the likelihood framework. Reed and Yu [16] identify outliers by using the so-called RX detector. Such an indicator measures the location of multivariate data points in the dispersion ellipsoid by means of the Mahalanobis distance. The authors assume that both ordinary and outlier data follow a multivariate normal distribution with same covariance matrix but different mean vectors.

* Corresponding author.
E-mail addresses: francesco.cesarone@uniroma3.it (F. Cesarone), rosella.giacometti@unibg.it (R. Giacometti), jacopomaria.ricci@uniroma3.it (J.M. Ricci).

https://doi.org/10.1016/j.cam.2024.116457
Received 12 April 2024; Received in revised form 18 October 2024
Available online 24 December 2024

Note that, even though Reed and Yu [16] assume the Gaussian distribution, other authors used the RX detector, removing such a strong distributional hypothesis. For example, Das and Sinha [17], Sinha [18] extend the work by Schwager and Margolin [11] to (nonnormal) elliptically symmetric distributions, and use the Mahalanobis distance as a detector. Since outliers affect both location and scale parameters, one way to overcome this issue is to use robust estimates. For example, Maronna [19] proposes robust M-estimators of multivariate mean vectors and covariance matrices. Both Stahel [20],Donoho [21] define, in their respective works, an affine equivariant robust estimator for location and scatter. Another popular method consists in computing the minimal covariance determinant (see e.g. [22–24]). Several authors propose alternative robust approaches adopting the comedian, a measure introduced by Falk [25], which generalizes the median absolute deviation. For instance, Sajesh and Srinivasan [26] use a method based on the comedian to detect anomalies for high dimensional data (see also [27]). Kazempour et al. [28] try to overcome the lack of robustness of PCA by replacing the covariance matrix with the comedian one. Finally, Cabana et al. [29] propose a number of shrinkage estimators based on the comedian to define a robust Mahalanobis distance with the aim of detecting outliers in multivariate data.

Especially when dealing with high-dimensional multivariate data, many techniques aim to find outliers in the univariate projections of such data to reduce the computational effort. For this reason, several studies have been devoted to identifying the "best" directions in which data must be projected to represent the variability of data most effectively. In Peña and Prieto [30,31], these directions correspond to those that maximize and minimize the kurtosis of the projected data. Domino [32] generalizes the Prieto and Peña's approach by choosing the direction that maximizes the fourth cumulant. Following this stream of literature, our work aims at detecting outliers, by projecting the data onto the direction that maximizes the Cumulant Generating Function (CGF).

In this paper, we refine some theoretical results of the methodologies proposed by Bernacchia and Naveau [33], Bernacchia et al. [34]. More precisely, we prove that CGF is a convex function, and, then, we characterize the CGF maximization problem on the unit n-circle as a concave minimization problem. Then, we extend the outlier detection methodology, based on the projections of multivariate data on the directions obtained by the classical PCA technique, to the directions that maximize CGF. Finally, we perform an extensive empirical analysis both on simulated and historical data, comparing our method with those described in [30–32].

The paper is organized as follows. In Section 2, we first introduce some preliminary concepts about the moment generating function, the cumulant generating function, and cumulants for univariate and multivariate random variables. Then, we report and refine some results developed in [33,34] for a generalization of the principal component analysis (PCA) technique. In Section 3, we describe our outlier detection algorithm, and the methods used to comparative purposes. For validation and testing, in Section 4, we present a thorough comparison of these methodologies both on artificial and real-world data. The real-world data application identifies the outliers as the materialization of financial crises. Finally, Section 5 contains some concluding remarks.

2. Theoretical framework

For the sake of completeness and readability, we recall below some notions about the moment generating function, the cumulant generating function, and the cumulants in the case both of a univariate and a multivariate random variable. Furthermore, we also report and refine some concepts introduced by Bernacchia and Naveau [33], Bernacchia et al. [34] for a generalization of the principal component analysis (PCA) technique.

Let X be a univariate random variable. If $\mathbb{E}[e^{\xi X}]$ exists and is finite $\forall \xi \in \mathbb{R}$, then the moment generating function of X is defined as follows

$$M_X(\xi) = \mathbb{E}[e^{\xi X}] = \sum_{m=0}^{+\infty} \mu_m \frac{\xi^m}{m!} \tag{1}$$

where $\mu_m = \mathbb{E}[X^m]$ is the mth raw moment of X, that can be obtained by differentiating m times w.r.t. ξ and setting $\xi = 0$, namely $\mu_m = M_X^{(m)}(0)$. The cumulant generating function (CGF) of X can be expressed as the logarithm of (1)

$$G_X(\xi) = \ln \mathbb{E}[e^{\xi X}] = \sum_{m=1}^{+\infty} k_m \frac{\xi^m}{m!}, \tag{2}$$

where k_m is the mth order cumulant of X. It is straightforward to see that k_m can be obtained by differentiating Expression (2) m times and setting $\xi = 0$, namely $k_m = G_X^{(m)}(0)$. Furthermore, the cumulants can be expressed as functions of the moments. For instance, the first 4 cumulants of X are as follows

$$\begin{aligned} k_1 &= \mathbb{E}[X] \\ k_2 &= \mathbb{E}[(X - \mathbb{E}[X])^2] \\ k_3 &= \mathbb{E}[(X - \mathbb{E}[X])^3] \\ k_4 &= \mathbb{E}[(X - \mathbb{E}[X])^4] - 3(\mathbb{E}[(X - \mathbb{E}[X])^2])^2 \end{aligned}$$

Note that the first two cumulants correspond to the mean and variance respectively, the third coincides with the third central moment, while from the fourth onward the cumulants are polynomial functions of the central moments with integer coefficients. From Expression (2), we also observe that for small values of ξ the cumulant generating function of X is essentially determined by its variance (if $k_1 = 0$), whereas for large ξ the contributions of the higher order cumulants become dominant. Furthermore, if $X \sim N(\mu, \sigma^2)$, then $G_X(\xi) = \mu\xi + \sigma^2 \frac{\xi^2}{2}$ since $k_m = 0$ for $m > 2$.

In the case of a multivariate random variable $\boldsymbol{X} = (X_1, \dots, X_n)$, its CGF becomes

$$G_{\boldsymbol{X}}(\boldsymbol{\xi}) = \ln \mathbb{E}[e^{\boldsymbol{\xi}^T \boldsymbol{X}}], \tag{3}$$

where $\boldsymbol{\xi} = (\xi_1, \dots, \xi_n) \in \mathbb{R}^n$. Denoting by $r = \|\boldsymbol{\xi}\|_2$ the euclidian norm of $\boldsymbol{\xi}$ and by $\boldsymbol{\theta}$ the versor of $\boldsymbol{\xi}$, we can write $\boldsymbol{\xi} = r\boldsymbol{\theta}$, and therefore the cumulant generating function of $\boldsymbol{X}$ can be expressed as follows

$$G_{\boldsymbol{X}}(r, \boldsymbol{\theta}) = \ln \mathbb{E}[e^{r\boldsymbol{\theta}^T \boldsymbol{X}}] = \sum_{m=1}^{+\infty} k_m(\boldsymbol{\theta}) \frac{r^m}{m!}. \tag{4}$$

where $k_m(\boldsymbol{\theta}) = \left. \frac{d^m G_{\boldsymbol{X}}(r, \boldsymbol{\theta})}{dr^m} \right|_{r=0}$ is the mth cumulant of $\boldsymbol{X}$ projected along the direction $\boldsymbol{\theta}$.

Note that the length r of $\boldsymbol{\xi}$ plays the same rule of ξ in the univariate case, namely if r is small then the information on $\boldsymbol{X}$ through CGF is basically represented by the covariance matrix of $\boldsymbol{X}$ (if data are centered, i.e., $\mathbb{E}[X_j] = 0$ with $j = 1, \dots, n$). Whereas if r is large then the information about $\boldsymbol{X}$ described by its CGF predominantly depends on the higher-order cumulants.

2.1. PCA through CGF

Following the work of Bernacchia and Naveau [33], we look for identifying the largest principal components, namely the directions that provide most of the variability present in the original data, by exploiting the information contained in the Cumulant Generating Function (CGF).

Using the classical PCA technique the principal component with the largest eigenvalue is the versor maximizing the variance, which, as we will show in the next section, also coincides with the optimal versor maximizing CGF (4) in the case of multivariate normal random variables, or in the case of generic random vectors but for small values of r. As mentioned by Bernacchia and Naveau [33], the CGF maximization aims to find the directions with the largest variability of the multivariate distribution not only through the first two cumulants, but also through the higher-order ones, which become dominant especially for data points that deviate widely from the mean of the random vector. Hence, for a fixed r, our goal is to find the directions that maximize (4), i.e.

$$\begin{aligned} \max_{\boldsymbol{\theta}} \quad & G_{\boldsymbol{X}}(r, \boldsymbol{\theta}) \\ \text{s.t.} \quad & \\ & \boldsymbol{\theta}^T \boldsymbol{\theta} = 1 \end{aligned} \tag{5}$$

As we will show in Section 2.3, Problem (5) is a constrained concave programming problem, where the main difficulty lies in the fact that concave problems normally have many local maxima points. In the next section, we present the CGF maximization approach on two artificial multivariate random vectors, normal and skew-normal.

2.2. Some special cases: multivariate normal and skew-normal

To better understand the rationale behind the approach of Bernacchia and Naveau [33], we report the CGF maximization procedure in the case of multivariate normal (Section 2.2.1) and skew-normal (Section 2.2.2) random variables, while in Section 2.3 we address the general case. We show that, in the case of a multivariate normal random vector, the optimal versor maximizing CGF collapses to the first principal component of the standard PCA technique. For a skew-normal, we will see that the optimal solution obtained from Problem (5) tends to the first principal component when the radius of the hypersphere is small, while, in the case of a large radius, the direction of the optimal versor deviates from the first component toward the direction where the multivariate distribution presents a fatter tail.

Without loss of generality, we henceforth assume that the data are centered around their mean.[1]

2.2.1. Multivariate normal case

Let $\boldsymbol{X} \sim N(\mathbf{0}, \boldsymbol{\Sigma})$, where, therefore, $\boldsymbol{X}$ is mean-centered and $\boldsymbol{\Sigma}$ is its covariance matrix. Thus, the cumulant generating function of $\boldsymbol{X}$ is

$$G_{\boldsymbol{X}}(\xi) = \frac{1}{2} \boldsymbol{\xi}^T \boldsymbol{\Sigma} \boldsymbol{\xi}.$$

Indeed, similar to the univariate case, for normal random vectors, the first two cumulants are exactly the mean and variance, while the higher-order cumulants are all equal to zero. Then, Problem (5) can be rewritten as

$$\begin{aligned} \max_{\boldsymbol{\theta}} \quad & \frac{r^2}{2} \boldsymbol{\theta}^T \boldsymbol{\Sigma} \boldsymbol{\theta} \\ \text{s.t.} \quad & \\ & \boldsymbol{\theta}^T \boldsymbol{\theta} = 1 \end{aligned} \tag{6}$$

This problem can be easily solved by means of the Lagrange multipliers method, where the Lagrangian function is $L(\lambda, \boldsymbol{\theta}) = \frac{r^2}{2} \boldsymbol{\theta}^T \boldsymbol{\Sigma} \boldsymbol{\theta} - \lambda(\boldsymbol{\theta}^T \boldsymbol{\theta} - 1)$. Hence, the corresponding first order conditions are $\nabla_\lambda L(\lambda, \boldsymbol{\theta}) = \boldsymbol{\theta}^T \boldsymbol{\theta} - 1 = 0$ and $\nabla_{\boldsymbol{\theta}} L(\lambda, \boldsymbol{\theta}) = r^2 \boldsymbol{\Sigma} \boldsymbol{\theta} - 2\lambda \boldsymbol{\theta} = 0$. Problem (6) has two optimal solutions, the versors $\hat{\boldsymbol{\theta}}_{\mathbf{1}}$ and $\hat{\boldsymbol{\theta}}_{\mathbf{2}} = -\hat{\boldsymbol{\theta}}_{\mathbf{1}}$, which are both eigenvectors of $\boldsymbol{\Sigma}$. Note that applying the classical PCA technique, the first principal components can be obtained by solving the following problem

$$\begin{aligned} \max_{\boldsymbol{\theta}} \quad & \boldsymbol{\theta}^T \boldsymbol{\Sigma} \boldsymbol{\theta} \\ \text{s.t.} \quad & \\ & \boldsymbol{\theta}^T \boldsymbol{\theta} = 1 \end{aligned} \tag{7}$$

[1] Therefore, for future research, it would be interesting to investigate centering techniques based on robust location estimators.

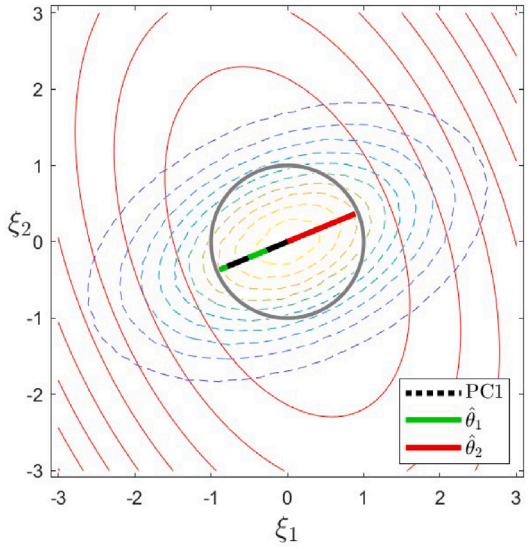


Fig. 1. Bivariate normal random variable: the dashed lines represent its isoprobability contours of $\boldsymbol{X}$, the red solid lines are the isocurves of $G_{\boldsymbol{X}}(r,\theta) = \frac{1}{2}\xi'\boldsymbol{\Sigma}\xi$, and the gray solid line is the circle of radius 1.

Since Problems (6) and (7) are equivalent except for the parameter $\frac{r^2}{2}$ in the objective function of (6), these problems lead to the same optimal solutions.

In Fig. 1, we give a graphical representation of the CGF maximization procedure in the case of a bivariate normal random variable. The red solid lines are the isocurves of $G_{\boldsymbol{X}}(\boldsymbol{\xi}) = \frac{1}{2}\boldsymbol{\xi}^T\boldsymbol{\Sigma}\boldsymbol{\xi}$, the gray solid line is the ball of radius 1, and the dashed lines represent the isoprobability contours of $\boldsymbol{X} \sim N(\mathbf{0}, \boldsymbol{\Sigma})$. The points, where the (red solid) isocurves and the (gray solid) circle of radius 1 are tangent, represent the optimal solutions of Problem (6), which coincide (except for a minus sign) with the first principal component lying on the major axis of the (dashed) local-dispersion ellipsoids, i.e., the direction of maximum variability of the bivariate random variable. We point out that the fact that $\hat{\boldsymbol{\theta}}_{\mathbf{1}}$ and $\hat{\boldsymbol{\theta}}_{\mathbf{2}}$ lie on the same direction is due to the symmetry of the bivariate normal distribution. Indeed, as we will see in the next section, in the case of a non-symmetric distribution $\hat{\boldsymbol{\theta}}_{\mathbf{1}}$ and $\hat{\boldsymbol{\theta}}_{\mathbf{2}}$ are in general on different directions, and these directions depend on r. Clearly, this phenomenon will be more evident for large r and for highly skewed distributions.

2.2.2. Multivariate skew-normal case

We report here the implementation of the CGF maximization procedure in the case of a multivariate skew-normal random variable.

Let $\boldsymbol{X} \sim SN(\boldsymbol{\eta}, \boldsymbol{\Sigma}, \boldsymbol{\alpha})$, where $\boldsymbol{\eta}$, $\boldsymbol{\Sigma}$ and $\boldsymbol{\alpha}$ are the location, the scale and the shape parameters, respectively. As discussed in [35], the probability density function (pdf) of $\boldsymbol{X}$ is

$$f_{\boldsymbol{X}}(\boldsymbol{x}) = 2\phi_n(\boldsymbol{x}-\boldsymbol{\eta};\boldsymbol{\Sigma})\Phi(\boldsymbol{\alpha}^T \operatorname{diag}(\boldsymbol{\sigma})^{-1}(\boldsymbol{x}-\boldsymbol{\eta})) \quad \text{with} \quad \boldsymbol{x} \in \mathbb{R}^n, \tag{8}$$

where $\phi_n(\boldsymbol{x}-\boldsymbol{\eta};\boldsymbol{\Sigma})$ is the pdf of an n-variate Gaussian random variable with mean $\boldsymbol{\eta}$ and covariance matrix $\boldsymbol{\Sigma}$, $\Phi(\cdot)$ is the cumulative distribution function (cdf) of a univariate standard normal random variable, $\boldsymbol{\alpha}$ is the degree of skewness (when $\boldsymbol{\alpha} = \mathbf{0}$, (8) collapses to the normal case), and $\operatorname{diag}(\boldsymbol{\sigma}) = \operatorname{diag}(\sigma_1, \ldots, \sigma_n)$, i.e., it represents the diagonal matrix of the standard deviations. As shown in [36,37], we have

$$\boldsymbol{\mu}_{\boldsymbol{X}} = \mathbb{E}[\boldsymbol{X}] = \boldsymbol{\eta} + \operatorname{diag}(\boldsymbol{\sigma})\boldsymbol{\delta} \tag{9}$$

$$\boldsymbol{\Sigma}_{\boldsymbol{X}} = \mathbb{V}\mathrm{ar}[\boldsymbol{X}] = \boldsymbol{\Sigma} - \operatorname{diag}(\boldsymbol{\sigma})\boldsymbol{\delta}\boldsymbol{\delta}^T \operatorname{diag}(\boldsymbol{\sigma}) \tag{10}$$

$$G_{\boldsymbol{X}}(\boldsymbol{\xi}) = \ln \mathbb{E}[e^{\boldsymbol{\xi}^T \boldsymbol{X}}] = \boldsymbol{\xi}^T\boldsymbol{\eta} + \frac{1}{2}\boldsymbol{\xi}^T\boldsymbol{\Sigma}\boldsymbol{\xi} + \ln[2\Phi(\sqrt{\frac{\pi}{2}}\boldsymbol{\delta}^T \operatorname{diag}(\boldsymbol{\sigma})\boldsymbol{\xi})], \tag{11}$$

where $\boldsymbol{\delta} = \dfrac{1}{\sqrt{\frac{\pi}{2}(1+\boldsymbol{\alpha}^T\boldsymbol{C}\boldsymbol{\alpha})}}\boldsymbol{C}\boldsymbol{\alpha}$, and $\boldsymbol{C} = \operatorname{diag}(\boldsymbol{\sigma})^{-1}\boldsymbol{\Sigma}\operatorname{diag}(\boldsymbol{\sigma})^{-1}$.

Since in this framework we work with a centered variable around its mean $\overline{\boldsymbol{X}} = \boldsymbol{X} - \boldsymbol{\mu}_{\boldsymbol{X}}$, its CGF becomes

$$G_{\overline{\boldsymbol{X}}}(\boldsymbol{\xi}) = -\boldsymbol{\xi}^T\boldsymbol{\mu}_{\boldsymbol{X}} + \boldsymbol{\xi}^T\boldsymbol{\eta} + \frac{1}{2}\boldsymbol{\xi}^T\boldsymbol{\Sigma}\boldsymbol{\xi} + \ln[2\Phi(\sqrt{\frac{\pi}{2}}\boldsymbol{\delta}^T \operatorname{diag}(\boldsymbol{\sigma})\boldsymbol{\xi})]. \tag{12}$$

Exploiting Expression (9) and denoting $\hat{\boldsymbol{\mu}} = \boldsymbol{\mu}_{\boldsymbol{X}} - \boldsymbol{\eta}$ and $\boldsymbol{\xi} = r\boldsymbol{\theta}$, we can write

$$G_{\overline{\boldsymbol{X}}}(r,\boldsymbol{\theta}) = -r\boldsymbol{\theta}^T\hat{\boldsymbol{\mu}} + \frac{r^2}{2}\boldsymbol{\theta}^T\boldsymbol{\Sigma}\boldsymbol{\theta} + \ln[2\Phi(\sqrt{\frac{\pi}{2}}r\hat{\boldsymbol{\mu}}^T\boldsymbol{\theta})], \tag{13}$$

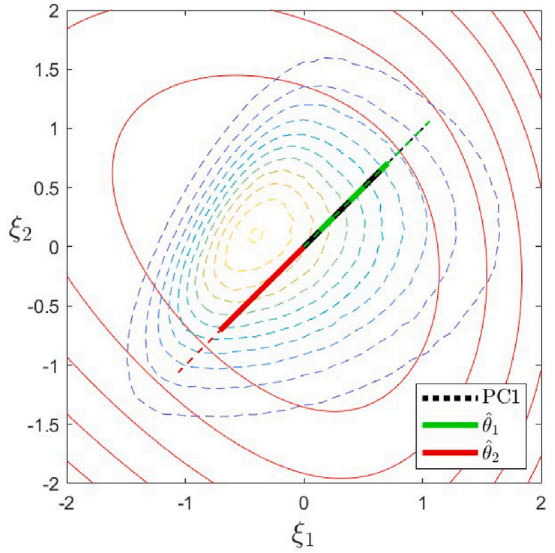


Fig. 2. Bivariate skew-normal random variable: the dashed lines represent its isoprobability contours, the red solid lines are the isocurves of $G_{\boldsymbol{X}}(r,\boldsymbol{\theta}) = -r\boldsymbol{\mu}^T\boldsymbol{\theta} + \frac{r}{2}\boldsymbol{\theta}^T\boldsymbol{\Sigma}\boldsymbol{\theta} + \ln[2\Phi(\sqrt{\frac{\pi}{2}}r\boldsymbol{\mu}^T\boldsymbol{\theta})]$, for small r.

For small r, using the Taylor expansion of $\ln(1+z) = z - \frac{z^2}{2} + \cdots$ and $2\Phi(z) = 1 + \operatorname{erf}(\frac{z}{\sqrt{2}}) = 1 + \sqrt{\frac{2}{\pi}}z + \cdots$, we have $\ln[2\Phi(\sqrt{\frac{\pi}{2}}r\hat{\boldsymbol{\mu}}^T\boldsymbol{\theta})] = \ln(1+\operatorname{erf}(\sqrt{\frac{\pi}{2}}r\hat{\boldsymbol{\mu}}^T\boldsymbol{\theta})) \simeq \ln(1+\sqrt{\frac{2}{\pi}}\sqrt{\frac{\pi}{2}}r\hat{\boldsymbol{\mu}}^T\boldsymbol{\theta}) \simeq r\hat{\boldsymbol{\mu}}^T\boldsymbol{\theta} - \frac{1}{2}r^2\boldsymbol{\theta}^T\hat{\boldsymbol{\mu}}\hat{\boldsymbol{\mu}}^T\boldsymbol{\theta}$. Therefore, Expression (13) can be approximated as follows

$$G_{\overline{\boldsymbol{X}}}(r,\boldsymbol{\theta}) \simeq \frac{r^2}{2}\boldsymbol{\theta}^T\left(\boldsymbol{\Sigma} - \hat{\boldsymbol{\mu}}\hat{\boldsymbol{\mu}}^T\right)\boldsymbol{\theta}, \tag{14}$$

where $\boldsymbol{\Sigma} - \hat{\boldsymbol{\mu}}\hat{\boldsymbol{\mu}}^T$ is exactly the covariance matrix of the skew-normal distribution $\boldsymbol{\Sigma}_{\boldsymbol{X}}$ as in (10). Hence, for small r, the optimal solution obtained from Problem (5) collapses to the first principal component, namely $\hat{\boldsymbol{\theta}}_{\mathbf{1}} = PC1$ and $\hat{\boldsymbol{\theta}}_{\mathbf{2}} = -PC1$. Fig. 2 exhibits a numerical example of the CGF maximization procedure in the case of a bivariate skew-normal random variable, when r is small.

For large r, we can examine the behavior of Eq. (13) using the asymptotic expansion of the error function (see, e.g., [38]) and distinguishing the case where $\hat{\boldsymbol{\mu}}^T\boldsymbol{\theta} \geq 0$ and $\hat{\boldsymbol{\mu}}^T\boldsymbol{\theta} < 0$. For this aim, we consider the following asymptotic expansion of the complementary error function for large real z

$$\operatorname{erfc}(z) = \frac{e^{-z^2}}{z\sqrt{\pi}}\sum_{n=0}^{\infty}(-1)^n\frac{(2n-1)!!}{(2z^2)^n} = \frac{e^{-z^2}}{z\sqrt{\pi}}\left(1 - \frac{1}{2z^2} + \frac{3}{4z^4} + \cdots\right) \tag{15}$$

where $(2n-1)!! = (2n-1)\cdot(2n-3)\cdots 3\cdot 1$. Therefore, for $z \to +\infty$

$$\operatorname{erf}(z) = 1 - \operatorname{erfc}(z) \simeq 1 - \frac{e^{-z^2}}{z\sqrt{\pi}} + \frac{e^{-z^2}}{2z^3\sqrt{\pi}} \tag{16}$$

This means that

$$2\Phi(z) = 1 + \operatorname{erf}\left(\frac{z}{\sqrt{2}}\right) \xrightarrow[z\to+\infty]{} 2 \tag{17}$$

Similarly, we obtain

$$2\Phi(-z) = 1 + \operatorname{erf}\left(\frac{-z}{\sqrt{2}}\right) = \operatorname{erfc}\left(\frac{-z}{\sqrt{2}}\right) \xrightarrow[z\to+\infty]{} -\frac{\sqrt{2}e^{-\frac{z^2}{2}}}{z\sqrt{\pi}} \tag{18}$$

$$\ln[2\Phi(-z)] = \ln\left[1 + \operatorname{erf}\left(\frac{-z}{\sqrt{2}}\right)\right] = \ln\left[\operatorname{erfc}\left(\frac{-z}{\sqrt{2}}\right)\right] \simeq \ln\left[-\frac{\sqrt{2}e^{-\frac{z^2}{2}}}{z\sqrt{\pi}}\right] \simeq -\frac{z^2}{2} \tag{19}$$

Then, examining Eq. (13) for $\hat{\boldsymbol{\mu}}^T\boldsymbol{\theta} \geq 0$ and large r, we have

$$G_{\overline{\boldsymbol{X}}}(r,\boldsymbol{\theta}) \simeq \frac{r^2}{2}\boldsymbol{\theta}^T\boldsymbol{\Sigma}\boldsymbol{\theta}, \tag{20}$$

while, for $\hat{\boldsymbol{\mu}}^T\boldsymbol{\theta} < 0$ and large r, we can write

$$G_{\overline{\boldsymbol{X}}}(r,\boldsymbol{\theta}) \simeq -r\boldsymbol{\theta}^T\hat{\boldsymbol{\mu}} + \frac{r^2}{2}\boldsymbol{\theta}^T\boldsymbol{\Sigma}\boldsymbol{\theta} - \frac{1}{2}\frac{\pi}{2}r^2\boldsymbol{\theta}^T\hat{\boldsymbol{\mu}}\hat{\boldsymbol{\mu}}^T\boldsymbol{\theta} \simeq \frac{r^2}{2}\boldsymbol{\theta}^T(\boldsymbol{\Sigma} - \frac{\pi}{2}\hat{\boldsymbol{\mu}}^T\hat{\boldsymbol{\mu}})\boldsymbol{\theta} \tag{21}$$

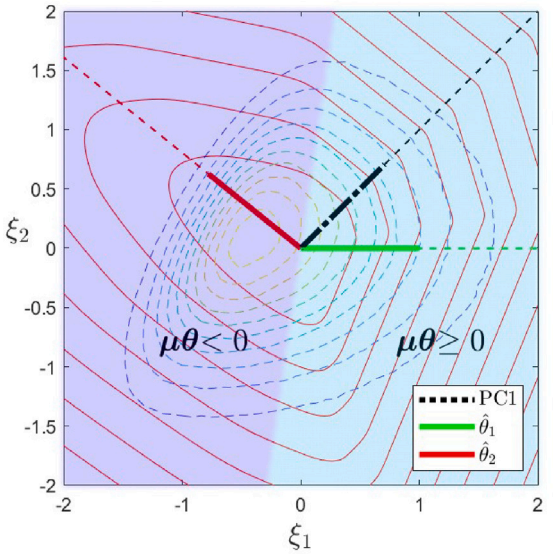


Fig. 3. Bivariate skew-normal random variable: the dashed lines represent its isoprobability contours, the red solid lines are the isocurves of $G_{\boldsymbol{X}}(r,\boldsymbol{\theta}) = -r\boldsymbol{\mu}^T\boldsymbol{\theta} + \frac{r}{2}\boldsymbol{\theta}^T\boldsymbol{\Sigma}\boldsymbol{\theta} + \ln[2\Phi(\sqrt{\frac{\pi}{2}}r\boldsymbol{\mu}^T\boldsymbol{\theta})]$, for large r.

In Fig. 3, we show a graphical representation of the CGF maximization procedure in the case of a bivariate skew-normal random variable. The red solid lines are the isocurves of $G_{\boldsymbol{X}}(r,\boldsymbol{\theta})$, and the dashed lines represent the isoprobability contours of $\boldsymbol{X} \sim SN(\mathbf{0}, \boldsymbol{\Sigma})$, where the diagonal entries of $\boldsymbol{\Sigma}$ are 1.2 and 0.5143, while the other entries are equal to 0, and $\boldsymbol{\alpha} = (4.365, -1.455)$.

2.3. Maximizing the non-parametric Cumulant Generating Function

In this section we address Problem (5) in the case of a generic discrete multivariate random variable $\boldsymbol{X}_t = (X_{1,t}, \dots, X_{n,t})$ defined on a discrete state space, where we assume T states of nature, each with probability π_t with $t = 1, \dots, T$. Therefore, to find the optimal direction $\widehat{\boldsymbol{\theta}}$ that maximizes the non-parametric Cumulant Generating Function, we solve the following optimization problem

$$\begin{aligned} \max_{\boldsymbol{\theta}} \quad & G_{\boldsymbol{X}}(r,\boldsymbol{\theta}) = \ln\left(\sum_{t=1}^{T} \pi_t e^{r\boldsymbol{\theta}^T\boldsymbol{X}_t}\right) \\ \text{s.t.} \quad & \\ & \boldsymbol{\theta}^T\boldsymbol{\theta} = 1 \end{aligned} \tag{22}$$

First, in the following theorem we show that the objective function of this problem is convex. Let us denote, for the sake of notation,

$$g(\boldsymbol{\xi}) = G_{\boldsymbol{X}}(\boldsymbol{\xi}) = \ln\left(\sum_{t=1}^{T} \pi_t e^{\boldsymbol{\xi}^T\boldsymbol{X}_t}\right) = G_{\boldsymbol{X}}(r,\boldsymbol{\theta}) = \ln\left(\sum_{t=1}^{T} \pi_t e^{r\boldsymbol{\theta}^T\boldsymbol{X}_t}\right), \tag{23}$$

where $\boldsymbol{\xi} = r\boldsymbol{\theta}$.

Theorem 1. *Let $g(\boldsymbol{\xi})$ be as in (23), where $\boldsymbol{\xi} \in \mathbb{R}^n$. Then, $g(\boldsymbol{\xi})$ is a convex function.*

Proof. Let $\boldsymbol{\xi}, \boldsymbol{\zeta} \in \mathbb{R}^n$, and $\gamma \in [0,1]$. Thus,

$$g(\gamma\boldsymbol{\xi} + (1-\gamma)\boldsymbol{\zeta}) = \ln\left(\sum_{t=1}^{T} \pi_t e^{(\gamma\boldsymbol{\xi}^T + (1-\gamma)\boldsymbol{\zeta}^T)\boldsymbol{X}_t}\right) = \ln\left(\sum_{t=1}^{T} \pi_t e^{\gamma\boldsymbol{\xi}^T\boldsymbol{X}_t}\ e^{(1-\gamma)\boldsymbol{\zeta}^T\boldsymbol{X}_t}\right)$$

Now, let $u_t = \pi_t e^{\boldsymbol{\xi}^T\boldsymbol{X}_t}$ and $v_t = \pi_t e^{\boldsymbol{\zeta}^T\boldsymbol{X}_t}\ \forall t = 1, \dots, T$, hence

$$g(\gamma\boldsymbol{\xi} + (1-\gamma)\boldsymbol{\zeta}) = \ln\left(\sum_{t=1}^{T} u_t^{\gamma} v_t^{1-\gamma}\right) \tag{24}$$

From Hölder's inequality, we have

$$\sum_{t=1}^{T} |x_t y_t| \leq \left(\sum_{t=1}^{T} |x_t|^p\right)^{\frac{1}{p}} \left(\sum_{t=1}^{T} |y_t|^q\right)^{\frac{1}{q}},$$

where $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^T$ and $p, q \in [1, \infty)$, with $\frac{1}{p} + \frac{1}{q} = 1$.

Table 1
Pseudocode of the CGF maximization procedure [44,45].

1. Generate N starting points (directions) $\theta_0^{(j)}$ with $j = 1, \dots, N$ belonging to the unit n-sphere (see [44]). Fix the step size $\delta_i = \delta = \frac{1}{r}$ and the tolerance ϵ sufficiently small (see [45]).
2. **for** $j = 1, \dots, N$
3. Set $i = 0$
4. **while** $\left\|\theta_{i+1}^{(j)} - \theta_i^{(j)}\right\| > \epsilon$
5. Compute the ascent direction as $\nabla_{\theta_i} G_{\boldsymbol{X}}(r, \theta_i^{(j)})$
6. Update $\theta_{i+1}^{(j)} = \theta_i^{(j)} + \delta_i \nabla_{\theta_i} G_{\boldsymbol{X}}(r, \theta_i^{(j)})$
7. Project by rescaling $\theta_{i+1}^{(j)}$ to unit norm, i.e., $\theta_{i+1}^{(j)} = \frac{\theta_{i+1}^{(j)}}{\left\|\theta_{i+1}^{(j)}\right\|}$
8. Update $i = i + 1$
9. **end while**
10. **end for**

Now, considering $x_t = u_t^{\gamma}$, $y_t = v_t^{1-\gamma}$, $\frac{1}{p} = \gamma$ and $\frac{1}{q} = 1 - \gamma$, we can apply Hölder's inequality to Expression (24) as follows

$$\begin{aligned} g(\gamma\boldsymbol{\xi} + (1-\gamma)\boldsymbol{\zeta}) = \ln\left(\sum_{t=1}^{T} u_t^{\gamma} v_t^{1-\gamma}\right) &\leq \ln\left(\left[\sum_{t=1}^{T} u_t^{\gamma \frac{1}{\gamma}}\right]^{\gamma}\left[\sum_{t=1}^{T} v_t^{(1-\gamma)\frac{1}{1-\gamma}}\right]^{1-\gamma}\right) = \\ &= \ln\left(\left[\sum_{t=1}^{T} u_t\right]^{\gamma}\left[\sum_{t=1}^{T} v_t\right]^{1-\gamma}\right) = \\ &= \gamma \ln\left(\sum_{t=1}^{T} u_t\right) + (1-\gamma)\ln\left(\sum_{t=1}^{T} v_t\right) = \\ &= \gamma g(\boldsymbol{\xi}) + (1-\gamma)\, g(\boldsymbol{\zeta}) \end{aligned}$$

which completes the proof. □

Therefore, Problem (22) consists in maximizing a convex function on an n-sphere, that is a nonconvex set. However, thanks to the following proposition we can relax such an n-sphere into an n-ball.

Proposition 2. *Problem (22) is equivalent to*

$$\begin{aligned} \max_{\theta} \quad & G_{\boldsymbol{X}}(r, \theta) = \ln\left(\sum_{t=1}^{T} \pi_t e^{r\theta^T \boldsymbol{X}_t}\right) \\ s.t. \quad & \\ & \theta^T \theta \leq 1 \end{aligned} \tag{25}$$

Proof. Since the n-ball $\theta^T\theta \leq 1$ can be seen as the convex hull of the n-sphere $\theta^T\theta = 1$, we can apply Theorem 32.2 of Rockafellar [39], which concludes the proof. □

Thus, Problem (25) is a concave programming problem, which is NP hard [40]. Indeed, the problem of globally maximizing a convex function on a convex set may have many local minima, hence finding the global maximum is a computationally difficult problem, and several approaches have been proposed in the literature to address this problem (see, e.g., [41,42] and references therein.

2.4. A heuristic for maximizing CGF

Similar to Bernacchia et al. [34], to solve Problem (25) we use a multistart method, which consists in generating several random initial points belonging to the n-sphere to determine a point in a neighborhood of a global maximum, and in using a local maximizer to efficiently determine a global maximum. We point out that we randomly generate initial points on the n-sphere, because, if there exists at least a global maximum of CGF on an n-ball, this belongs to its frontier (see Corollary 32.3.1 of [39]). Furthermore, since the constraint of Problem (25) defines a convex set, we use the projected gradient algorithm as the local maximizer (see [43]). In Table 1, we summarize the CGF maximization procedure and make its MATLAB code public in the web page https://www.francescocesarone.com/papers.

More precisely, in Expression (23), as in Peña and Prieto [30], Bernacchia et al. [34], and Domino [32], we assume to work with a sample coming from the data generating process, hence $\pi_t = \frac{1}{T}$ for $t = 1, \dots, T$, and, therefore, the CGF of a discrete multivariate variable $\boldsymbol{X}_t = (X_{1,t}, \dots, X_{n,t})$ becomes

$$G_{\boldsymbol{X}}(r, \boldsymbol{\theta}) = \ln \frac{1}{T} \sum_{t=1}^{T} e^{r\boldsymbol{\theta}^T \boldsymbol{X}_t}. \tag{26}$$

We point out that the proposed methodology works for any statistical assumption on the discrete probability distribution of $\boldsymbol{X}_t$, with $t = 1, \dots, T$.

For a fixed r, we set $N = 100$ and $\delta_i = \frac{1}{r}$, where N is the number of starting points of the multistart heuristic and δ_i is the step size of the projected gradient algorithm (see Table 1). Furthermore, from Expression (26), for a given r, the gradient of $G_{\boldsymbol{X}}(r, \boldsymbol{\theta})$ is

$$\nabla_{\boldsymbol{\theta}} G_{\boldsymbol{X}}(r, \boldsymbol{\theta}) = r \frac{\frac{1}{T} \sum_{t=1}^{T} \boldsymbol{X}_t e^{r\boldsymbol{\theta}^T \boldsymbol{X}_t}}{\frac{1}{T} \sum_{t=1}^{T} e^{r\boldsymbol{\theta}^T \boldsymbol{X}_t}} \tag{27}$$

Therefore, in Step 6 of the pseudocode in , the iterative scheme to find a local optimum is defined by

$$\boldsymbol{\theta}_{i+1} - \boldsymbol{\theta}_i = \delta_i \nabla_{\boldsymbol{\theta}_i} G_{\boldsymbol{X}}(r, \boldsymbol{\theta}_i) = \frac{\sum_{t=1}^{T} \boldsymbol{X}_t e^{r\boldsymbol{\theta}_i^T \boldsymbol{X}_t}}{\sum_{t=1}^{T} e^{r\boldsymbol{\theta}_i^T \boldsymbol{X}_t}}, \tag{28}$$

while, in Step 7, $\boldsymbol{\theta}_{i+1}$ is normalized to one.

Thus applying the CGF maximization procedure to Problem (25), we can obtain the local maxima $\hat{\boldsymbol{\theta}} = \hat{\boldsymbol{\theta}}(r)$, which, except for symmetric distributions (see Section 2.2.1), depend on r. As discussed in Section 2, if r (the distance between data points and the center) is small, then the CGF maximization procedure essentially picks the first principal component of the classical PCA. Whereas, if r is large, then the optimal directions $\hat{\boldsymbol{\theta}}$ maximizing $G_{\boldsymbol{X}}(r, \boldsymbol{\theta})$ strongly depend on the higher-order cumulants. On the other hand, higher values of r produce a less accurate estimate of the function $G(r, \boldsymbol{\theta})$, since the data sample is finite and a few outliers could heavily influence the higher-order cumulants. Therefore, when setting the value of r, one must consider the trade-off between finding $\hat{\boldsymbol{\theta}}(r)$ with large r (thus involving higher-order cumulants in the procedure) and obtaining an accurate estimate of (26). As suggested by Bernacchia et al. [34], we set a value $\bar{r}$ such that the CGF estimation error is limited. We identify such an error by means of the relative variance of the CGF estimate, ε_G^2, that, as shown in Appendix A, assuming i.i.d. normally distributed random vectors $\boldsymbol{X}_t \sim N_n(\boldsymbol{0}, \boldsymbol{\Sigma})$ $\forall t = 1, \dots, T$, is as follows

$$\varepsilon_G^2 = \frac{\mathbb{V}\mathrm{ar}[G]}{(\mathbb{E}[G])^2} \simeq \frac{4}{T} \frac{e^{\bar{r}^2 \lambda_1} - 1}{\bar{r}^4 (\lambda_1)^2}, \tag{29}$$

where T is the number of data points, and λ_1 is the largest eigenvalue computed by the standard PCA technique. In the experimental analysis, $\bar{r}$ is found through (29) by setting $\varepsilon_G = 10\%$.

3. Outlier detection methodologies

In this section, we introduce the proposed outlier detection algorithm for multivariate data, named the MaxCGF algorithm, consisting in finding outliers in univariate projections of such data. Our approach is compared with two other projection-based methods, i.e., one developed by Peña and Prieto [30,31] and the other developed by Domino [32]. Their main difference relies on the selected directions onto which the data are projected. More precisely, Peña and Prieto [30,31] consider the directions for which the kurtosis of the projected data shows the highest and lowest values. Domino [32] takes into account the projection directions for which a multivariate series exhibits the highest absolute value of the 4th order cumulant. We propose a more general approach that is based on the projection directions that maximize the Cumulant Generating Function of the multivariate data. In the following we report the main steps of our procedure.

1. Preprocess the original data $\boldsymbol{X}$, yielding centered data $\boldsymbol{Y}$. Note that while both Peña and Prieto [30,31], Domino [32] standardize $\boldsymbol{X}$, we only center them.[2]
2. Find the directions that maximize CGF of $\boldsymbol{Y}$.
3. Compute the univariate projection $\boldsymbol{Z}$ of $\boldsymbol{Y}$ in the directions identified in Step 2.
4. Remove outliers. Note that to identify whether a generic element z_t of $\boldsymbol{Z} = \{z_t\}_{t=1,\dots,T}$ is an outlier, similar to Peña and Prieto [30,31], Domino [32], we compute the following quantity

$$q_t = \frac{|z_t - \mathrm{median}(\boldsymbol{Z})|}{MAD(\boldsymbol{Z})}, \quad t = 1, \dots, T, \tag{30}$$

where T is the length of the time series, and $MAD(\boldsymbol{Z})$ represents the median absolute deviation of $\boldsymbol{Z}$. Then, a generic outcome z_t is classified as an outlier if its corresponding q_t exceeds a fixed threshold β. Such a threshold is selected to cover the whole detection range both in terms of True Positive Rate and in terms of False Positive Rate (for more details, see Section 4). In order to determine the optimal β, we consider equally spaced values of β ranging from 0.5 to 10 with a step of 0.25.

[2] Indeed, as explained in Bernacchia and Naveau [33], the focus of the procedure developed by the authors consists in finding large anomalies, whereas the standardization would identify independent components, making it a special case of Independent Component Analysis (see e.g. [46,47]).

Table 2
Pseudocode of the MaxCGF algorithm.

1. Fix the threshold β;
2. Center data, i.e., $\boldsymbol{Y}^{(0)} = \boldsymbol{X} - \boldsymbol{\mu}$;
3. Find the N directions $\hat{\boldsymbol{\theta}}_1^{(0)}, \hat{\boldsymbol{\theta}}_2^{(0)}, \ldots, \hat{\boldsymbol{\theta}}_N^{(0)}$, maximizing CGF of $\boldsymbol{Y}^{(0)}$ (see Table 1);
4. **for** $j = 1 : N$ (i.e., for each direction $\hat{\boldsymbol{\theta}}_j^{(0)}$)
5. Set $i = 0$;
6. Project $\boldsymbol{Y}^{(0)}$ on $\hat{\boldsymbol{\theta}}_j^{(0)}$, thus obtaining the vector $\boldsymbol{Z}_j^{(0)} = \boldsymbol{Y}^{(0)}\hat{\boldsymbol{\theta}}_j^{(0)}$;
7. Compute $\mathrm{Kur}_j^{(0)} = \mathbb{K}\mathrm{ur}(\boldsymbol{Z}_j^{(0)})$, i.e., the kurtosis of $\boldsymbol{Z}_j^{(0)}$;
8. Compute $\overline{\mathrm{K}}_0 = \frac{1}{N}\sqrt{\sum_{j=1}^{N}(\mathrm{Kur}_j^{(0)})^2}$, i.e., the mean squared kurtosis over the N directions;
9. **while** $\overline{\mathrm{K}}_i < \overline{\mathrm{K}}_{i-1}$ or $i = 0$
10. Compute the vector $\boldsymbol{q}_j = \{q_{j,t}\}_{t=1,\ldots,T}$ as in Eq. (30) for $\boldsymbol{Z}_j^{(i)}$;
11. Remove the outliers when $\max_{1\le j\le T} q_{j,t} > \beta$, thus obtaining $\boldsymbol{Y}^{(i+1)}$;
12. From $\boldsymbol{Y}^{(i+1)}$, compute $\hat{\boldsymbol{\theta}}_j^{(i+1)}$ by using the algorithm in Table 1;
13. Project $\boldsymbol{Y}^{(i+1)}$ on $\hat{\boldsymbol{\theta}}_j^{(i+1)}$, thus obtaining the vector $\boldsymbol{Z}_j^{(i+1)} = \boldsymbol{Y}^{(i+1)}\hat{\boldsymbol{\theta}}_j^{(i+1)}$;
14. Compute $\overline{\mathrm{K}}_{i+1}$, i.e., the mean squared kurtosis of $\boldsymbol{Z}_j^{(i+1)}$, with $j = 1, \ldots, N$;
15. Update $i = i + 1$
16. **end while**
17. $\boldsymbol{Y}^{(0)} = \boldsymbol{Y}^{(i+1)}$, $\hat{\boldsymbol{\theta}}_j^{(0)} = \hat{\boldsymbol{\theta}}_j^{(i+1)}$
18. **end for**

5. Repeat Steps 2, 3 and 4 until the mean squared kurtosis over the projections $\overline{\boldsymbol{Z}}$ increases (see [32]).

In Table 2, we summarize the MaxCGF algorithm (pseudocode) and make its MATLAB code public in the web page https://www.francescocesarone.com/papers.

In the next section, we compare the outlier detection ability of our method with that of two alternative methods proposed by Peña and Prieto [30,31], Domino [32].

4. Experimental analysis

We provide here a thorough empirical analysis, where the three outlier detection procedures discussed in Section 3 are tested and compared both using simulated data (see Section 4.1) and financial real-world data (see Section 4.2). Performances are evaluated using the Receiver Operating Characteristic (ROC) curves, which are built by plotting the True Positive Rate (TPR) vs. the False Positive Rate (FPR) of each methodology for different levels of the threshold β. TPR is the rate of truly detected outliers w.r.t. all possible outliers, and FPR (i.e., the type I error) is the rate of falsely detected outliers w.r.t. all the non-outlier data. Furthermore, we also provide two other performance measures related to the ROC curve, namely the Area Under the Curve (AUC) and Youden's J Statistic (YJS). AUC is the area underneath the ROC curve: if a method perfectly distinguishes between outlier and non-outlier data, then its AUC is equal to 1; if, on the other hand, AUC is less than or equal to 0.5, then this measure is uninformative, i.e., there is no difference in performance between the analyzed method and one that relies on random choices. YJS represents the difference between TPR and FPR, and again, the higher, the better. The Best Cutoff Value (BCV) is the largest YJS, and β^* is the threshold value corresponding to BCV. Although β^* is not a performance measure in the strict sense, it provides information about the practical use of algorithms and allows one to identify whether there is an optimal range of β for which algorithms work best.

Note that in our experiments we report the computational results for all the three methods only in three instances (standard normal, normal, and real-world data from the Dow Jones market). In the remaining cases, we report only the computational results of two methods since the Peña–Prieto algorithm does not provide reasonable results, and the generated ROC curves do not show a nondecreasing trend at some points, or yield AUC values less than 0.5.

We implemented all the experiments on a workstation with Intel(R) Xeon(R) CPU (E5-2623 v4, 2.6 GHz, 64 Gb RAM) under MS Windows 10, using MATLAB 9.12.0.

4.1. Computational results for simulated data

Artificial simulated returns are drawn from four different distributions: standard normal (Section 4.1.1), normal (Section 4.1.2), skew-normal (Section 4.1.3) and Student's t (Section 4.1.4), with different values of n (the number of marginals, i.e., assets) and T (the number of scenarios). We consider three instances, i.e., $n = 10$ and $T = 100$, $n = 20$ and $T = 250$, and $n = 30$ and $T = 500$. As for

Table 3
Performances for the standard normal dataset. We mark in green the best, in yellow the intermediate, and in red the worst results.

Method	AUC	BCV	Time (min.)	β^*
MaxCGF	0.9843	0.9533	0.6032	3.25
Domino	0.9316	0.7489	21.6774	4.25
Peña–Prieto	0.9867	0.9867	66.6850	2.25

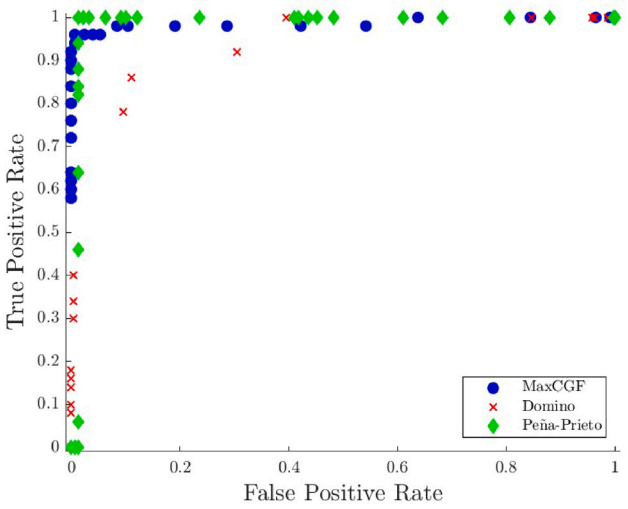


Fig. 4. ROC curves of the three algorithms for the standard normal market.

the outliers matrix $\boldsymbol{O}_{(T_{\text{out}}\times n_{\text{out}})}$, we set its size to $T_{\text{out}} \times n_{\text{out}}$. Specifically, we examine the cases with $n_{\text{out}} = 0.4\ n$ and $T_{\text{out}} = 0.05\ T$, $n_{\text{out}} = 0.5\ n$ and $T_{\text{out}} = 0.1\ T$, and $n_{\text{out}} = 0.6\ n$ and $T_{\text{out}} = 0.2\ T$. Below, we report the main steps for generating outliers and substituting them into the ordinary data.

1. Compute the sample covariance matrix $\boldsymbol{\Sigma}$ from a real-world dataset (here, the Dow Jones dataset from 07/2004 to 07/2006, see Section 4.2).
2. Generate the ordinary data $\boldsymbol{X}_{(T\times n)}$ with T scenarios from an n-dimensional random vector using $\boldsymbol{\Sigma}$ (in the standard normal case, $\boldsymbol{\Sigma} = \boldsymbol{I}$); generate the outlier data $\boldsymbol{O}_{(T_{\text{out}}\times n_{\text{out}})}$ using $\boldsymbol{\Sigma}$ multiplied by $c = \{2, 5, 10, 15\}$.
3. Substitute the elements of $\boldsymbol{O}_{(T_{\text{out}}\times n_{\text{out}})}$ in $\boldsymbol{X}_{(T\times n)}$. More precisely, randomly select n_{out} columns (with T_{out} elements) of $\boldsymbol{X}_{(T\times n)}$ and substitute them with the columns of $\boldsymbol{O}_{(T_{\text{out}}\times n_{\text{out}})}$.

For the sake of readability, in the following sections, we report and discuss only the cases where $n = 30$ and $T = 500$, $n_{\text{out}} = 0.5\ n$ and $T_{\text{out}} = 0.1\ T$, and $c = 15$. The remaining cases are provided in the online supplemental material. In Section 4.1.1, we report the standard normal case, while in Sections 4.1.2 (normal), 4.1.3 (skew-normal), and 4.1.4 (Student's t), we report the cases where the covariance matrix $\boldsymbol{\Sigma}$ is estimated from a real-world dataset, thus generating correlated random variables.

4.1.1. The standard normal random vector case

For this experimental case, the ordinary and outlier data are $\boldsymbol{X}_{(T\times n)} \sim N(\boldsymbol{0}, \boldsymbol{I})$ and $\boldsymbol{O}_{(0.1T\times 0.5n)} \sim N(\boldsymbol{0}, 15\boldsymbol{I})$.

In Table 3, we report the computational results obtained by using the three methods described in Section 3. More precisely, the Peña–Prieto method obtains an almost perfect score of 0.9867 both for the AUC and the BCV performance measures. The Domino algorithm shows the lowest performance w.r.t. the other two methods, both in terms of AUC (= 0.9316) and BCV (= 0.7489). The MaxCGF method almost achieves the highest values in terms of accuracy and clearly shows a significant advantage in terms of computational burden. Its running time is less than 1 min, compared with 15 min and more than 1 h spent by the Domino and Peña–Prieto methods, respectively.

Fig. 4 reports the ROC curves of the three methods, which, as already noted, show high outlier detection abilities in this experiment.

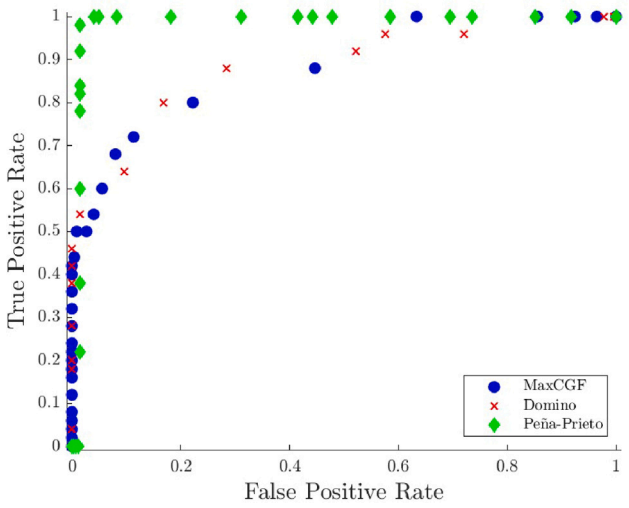


Fig. 5. ROC curves of the three algorithms for the normal market.

Table 4
Performances for the normal dataset.

Method	AUC	BCV	Time (min.)	β^*
MaxCGF	0.8811	0.6067	0.9592	8.00
Domino	0.8809	0.6311	14.3293	7.50
Peña–Prieto	0.9847	0.9644	67.2382	1.75

Table 5
Performances for the skew-normal dataset.

Method	AUC	BCV	Time (min.)	β^*
MaxCGF	0.9140	0.6911	1.0387	7.25
Domino	0.8896	0.6467	15.8041	6.50
Peña–Prieto	–	–	–	–

4.1.2. The normal random vector case

In this case, the ordinary and outlier data are $\boldsymbol{X}_{(T\times n)} \sim N(\boldsymbol{0}, \boldsymbol{\Sigma})$ and $\boldsymbol{O}_{(0.1T\times 0.5n)} \sim N(\boldsymbol{0}, 15\boldsymbol{\Sigma})$.

The empirical results of the ROC analysis are shown in Fig. 5 and in Table 4. Here, the ranking of the three algorithms is similar to that of the standard normal dataset. Again, the Peña–Prieto algorithm outperforms the other two both in terms of AUC and BCV, while the MaxCGF algorithm is the most efficient; however, the Domino algorithm yields the second best BCV. It is noteworthy that, compared to the standard normal case, the introduction of a correlation structure seems to worsen the results for the MaxCGF and Domino algorithms, while that of Peña–Prieto does not experience noticeable changes.

4.1.3. The skew-normal random vector case

For this experiment, the ordinary and outlier data are $\boldsymbol{X}_{(T\times n)} \sim SN(\boldsymbol{0}, \boldsymbol{\Sigma}, \boldsymbol{\alpha})$ and $\boldsymbol{O}_{(0.1T\times 0.5n)} \sim SN(\boldsymbol{0}, 15\boldsymbol{\Sigma}, \boldsymbol{\alpha})$, where $\boldsymbol{\alpha}$ is a vector whose elements are uniformly distributed in the interval $[-1, 4]$.

As shown in Table 5 and in Fig. 6, the MaxCGF algorithm provides, in the case of asymmetric distribution, the highest AUC (= 0.9140) and BCV (= 0.6467), and it is the most efficient approach. The Domino method also performs well, although lower than the MaxCGF method. Its AUC is 0.8896, and its BCV is 0.6467, while its running time is 15 min compared with 1 min spent by the MaxCGF method.

4.1.4. The Student's t random vector case

Here, the ordinary and outlier data follow an n-variate Student's t distribution, $\boldsymbol{X}_{(T\times n)} \sim St(\boldsymbol{0}, \boldsymbol{\Sigma}, \nu)$ and $\boldsymbol{O}_{(0.1T\times 0.5n)} \sim St(\boldsymbol{0}, 15\boldsymbol{\Sigma}, \nu)$, where ν is set to 5, 10, 100, 1000. Clearly, when ν is sufficiently high, the Student's t random vector approaches the normal one.

Before testing the three methods described in Section 3, we examine the outlier detection procedure by using the projection directions provided by the CGF maximization and by the classical PCA.

As shown in Table 6, the latter is much more efficient in terms of running time, but, except for $\nu = 1000$, the MaxCGF method is more precise, as shown by the values of AUC and BCV. Furthermore, the accuracy of the classical PCA tend to increase with ν, namely when the simulated data tend to have a normal distribution. Fig. 7 exhibits the ROC curves of such experiments.

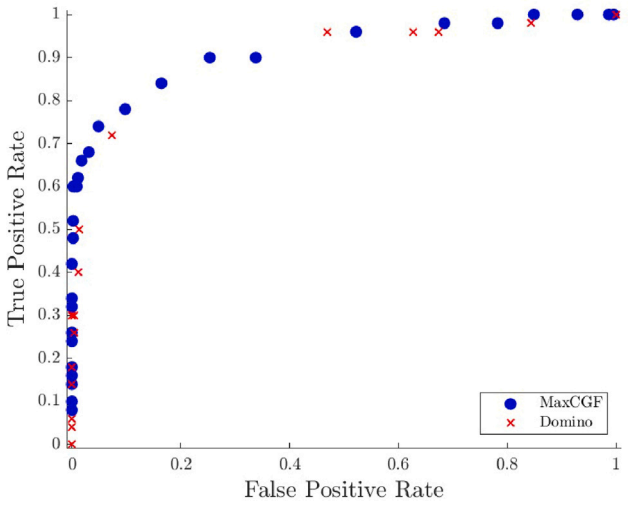


Fig. 6. ROC curves of the MaxCGF and Domino algorithms for the skew-normal market.

Table 6
Outlier detection performances using the CGF maximization and the classical PCA for the Student's t random vector.

ν	Method	AUC	BCV	Time (min.)
5	**Classical PCA**	0.7757	0.4467	0.8389
	MaxCGF	0.8544	0.5933	22.5747
10	**Classical PCA**	0.7988	0.4089	0.7955
	MaxCGF	0.8333	0.5356	10.0730
100	**Classical PCA**	0.8434	0.5600	0.7952
	MaxCGF	0.8525	0.6311	21.8163
1000	**Classical PCA**	0.8504	0.5756	0.7755
	MaxCGF	0.8409	0.5489	17.5729

Table 7
Performances for the Student's t dataset for $\nu = 10$.

Method	AUC	BCV	Time (min.)	β^*
MaxCGF	0.8333	0.5356	10.0730	7.25
Domino	0.8498	0.5622	8.6374	7.50
Peña–Prieto	–	–	–	–

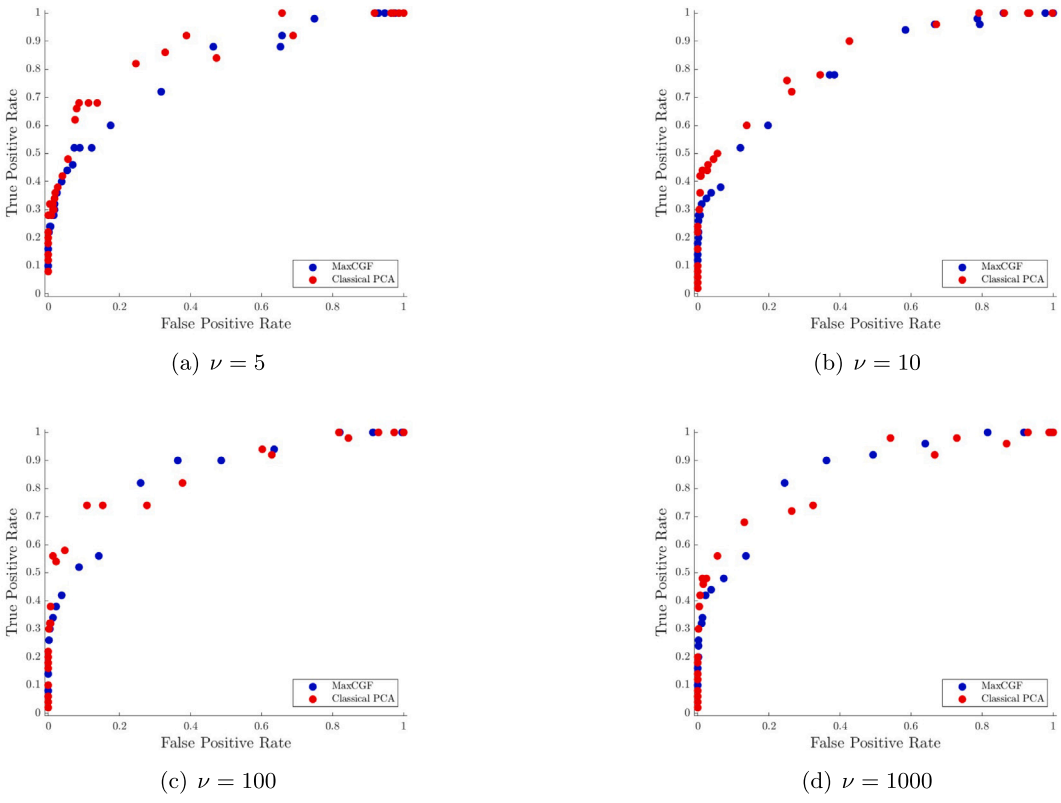
Table 8
Performances for the Student's t dataset for $\nu = 30$.

Method	AUC	BCV	Time (min.)	β^*
MaxCGF	0.9116	0.7044	8.0739	7.75
Domino	0.8903	0.6267	19.1567	7.25
Peña–Prieto	–	–	–	–

For these experiments, we also compare the performances of the Peña–Prieto, Domino, and MaxCGF methods considering the degrees of freedom of the Student's t random vector, $\nu = 10$ and $\nu = 30$. Note that, as mentioned at the beginning of Section 4, the Peña–Prieto algorithm does not provide reasonable results. Tables 7 and 8 show the values of AUC, BCV, and CPU time for $\nu = 10$ and $\nu = 30$, respectively. In the former case, the Domino approach slightly outperforms the MaxCGF one, while in the latter case, the MaxCGF method yields better results in terms of AUC, BCV, and running time. Fig. 8 exhibits the two methods' ROC curves, highlighting elevated outlier detection abilities in these experiments.

4.2. Computational results for financial real-world data

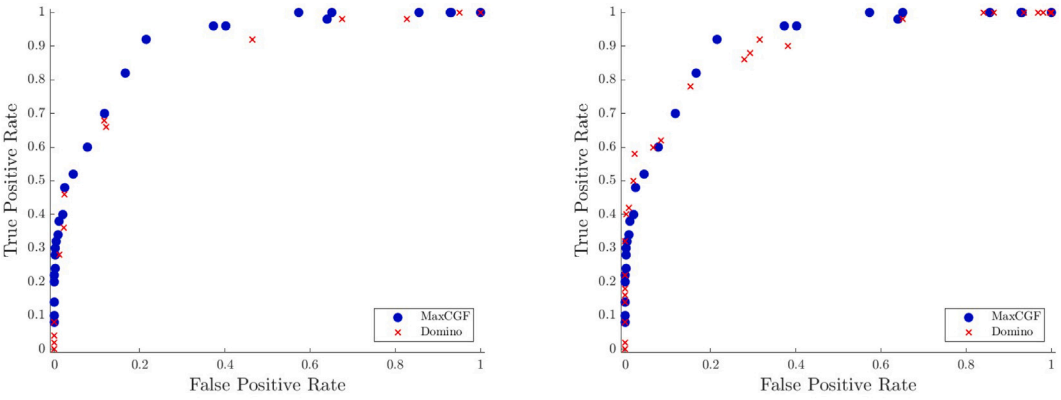
In this section, we use the three outlier detection methods to identify financial crises, which, as mentioned in the introduction, can be seen as periods when markets experience atypical behavior. More precisely, we apply the outlier detection analysis to six real-world financial datasets belonging to major stock markets across the world. In Table 9, we provide some details about these datasets, which consist of daily prices, adjusted for dividends and stock splits, obtained from the data provider *LESG Data & Analytics* (formerly



(a) $\nu = 5$ (b) $\nu = 10$

(c) $\nu = 100$ (d) $\nu = 1000$

Fig. 7. ROC curves using the CGF maximization and the classical PCA for the Student's t random vector.



(a) $\nu = 10$. (b) $\nu = 30$.

Fig. 8. ROC curves of the MaxCGF and Domino algorithms for the Student's t random vector.

known as *Refinitiv-Datastream*, see LSEG [48]). From prices, we use both linear and logarithmic returns for the empirical analysis. The time series of these daily asset returns are publicly available on the web page https://www.francescocesarone.com/data-sets. Since the results obtained are practically identical, we report here only those obtained by means of the linear returns. The data analyzed are daily returns from April 9, 2019, to March 23, 2020 (approximately one financial year). We have chosen this time horizon because it contains a period of high instability due to the COVID-19 pandemic, the effects of which occur approximately in early February 2020 (see, e.g. Shu et al. [49]).

In Tables 10–15, we report the computational results for all the datasets listed in Table 9. These tables show that AUC obtained by the MaxCGF approach is between 0.8021 and 0.9057, and is always higher than that achieved by the other two methods.

Table 9
List of the daily datasets analyzed.

Market	Abbrev.	# assets	Country	From-To
Dow Jones Industrial Average	DJIA	30	USA	09/04/2019-23/03/2020
Euro Stoxx 50	STOXX50	47	Eurozone	
DAX 30	DAX	28	Germany	
CAC 40	CAC	40	France	
FTSE 100	FTSE	98	UK	
EuroNext 100	N100	98	Eurozone	

Table 10
Performances for DJIA.

Method	AUC	BCV	Time (min.)	β^*
MaxCGF	0.9057	0.7156	5.9174	6.75
Domino	0.8282	0.6178	25.6869	5.75
Peña–Prieto	0.7873	0.5511	84.0113	6.50

Table 11
Performances for STOXX50.

Method	AUC	BCV	Time (min.)	β^*
MaxCGF	0.8312	0.5666	3.6558	8.00
Domino	0.7480	0.4713	92.4653	7.50
Peña–Prieto	–	–	–	–

Table 12
Performances for DAX.

Method	AUC	BCV	Time (min.)	β^*
MaxCGF	0.8866	0.6555	3.8149	5.75
Domino	0.7924	0.5701	17.2033	7.00
Peña–Prieto	–	–	–	–

Table 13
Performances for CAC.

Method	AUC	BCV	Time (min.)	β^*
MaxCGF	0.8326	0.5187	3.7390	8.25
Domino	0.7391	0.4637	75.1684	6.50
Peña–Prieto	–	–	–	–

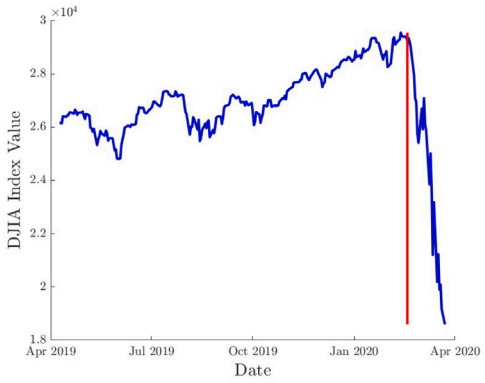
Table 14
Performances for FTSE.

Method	AUC	BCV	Time (min.)	β^*
MaxCGF	0.8021	0.5207	5.4309	7.50
Domino	0.7061	0.4641	2644.9669	7.50
Peña–Prieto	–	–	–	–

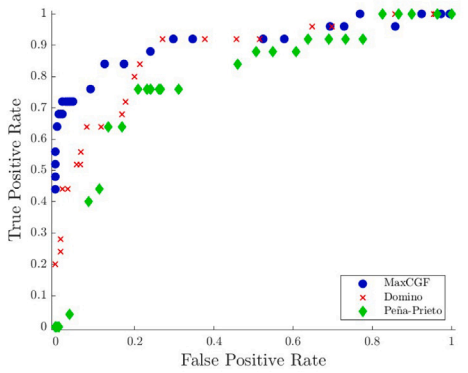
Table 15
Performances for N100.

Method	AUC	BCV	Time (min.)	β^*
MaxCGF	0.8643	0.5474	6.2306	7.75
Domino	0.6794	0.3624	2986.0548	7.75
Peña–Prieto	–	–	–	–

In Figs. 9–14, we also provide the ROC curves for all the financial market analyzed. Interestingly, TPR of the MaxCGF algorithm is high for values of FPR slightly higher than 0.1. This means that our algorithm is able to correctly detect a large portion of outliers misclassifying only few ordinary data (as outlier) even for high values of the threshold β. Furthermore, the MaxCGF method is always the least time-consuming, only taking a few minutes to complete the analysis for all the financial datasets analyzed. Conversely, both the Domino and the Peña–Prieto algorithms are heavily influenced by the number of assets n. Indeed, on the one hand, the Peña–Prieto procedure is able to produce results only for DJIA. On the other hand, for the Domino approach, the running time ranges from about 17 min for DAX ($n = 28$) to around two days for the N100 ($n = 98$). Summing up, as highlighted in the tables, the MaxCGF method always shows better results for all the examined performance measures, followed by the Domino method.

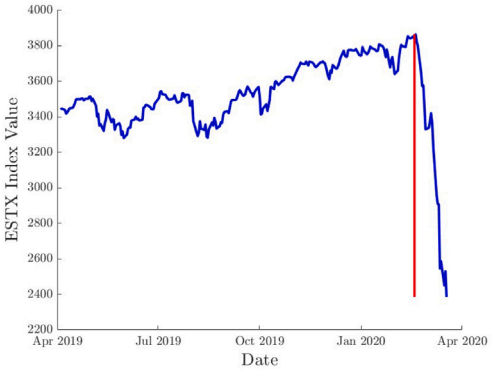


(a) Value of the Dow Jones index. The red line indicates the beginning of the crisis.

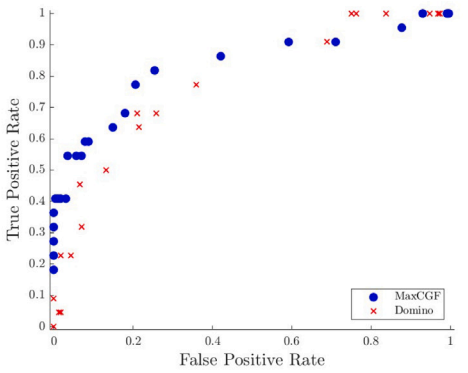


(b) ROC curves of the three algorithms for the DJIA dataset.

Fig. 9. DJIA dataset.

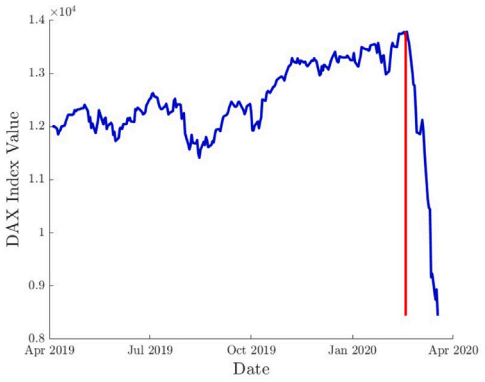


(a) Value of the Eurostoxx50 index. The red line indicates the beginning of the crisis.

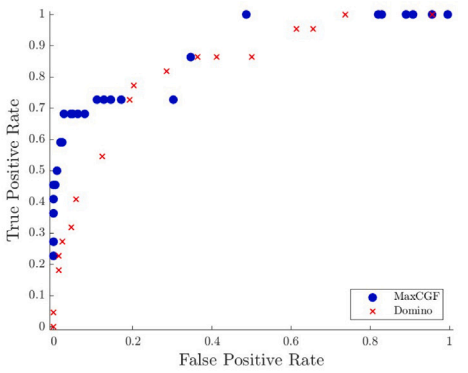


(b) ROC curves of the MaxCGF and Domino algorithms for the STOXX50 dataset.

Fig. 10. STOXX50 dataset.

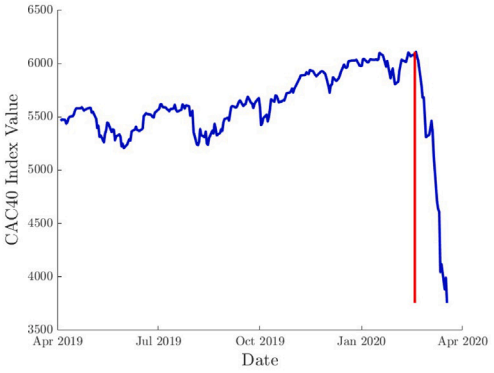


(a) Value of the DAX index. The red line indicates the beginning of the crisis.

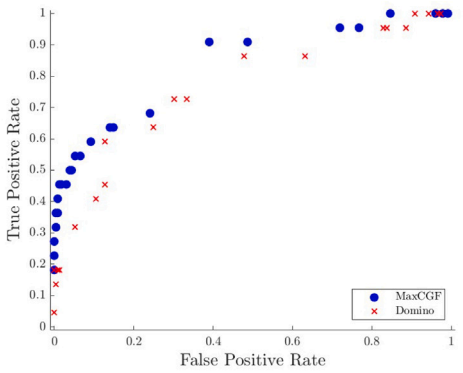


(b) ROC curves of the MaxCGF and Domino algorithms for the DAX dataset.

Fig. 11. DAX dataset.



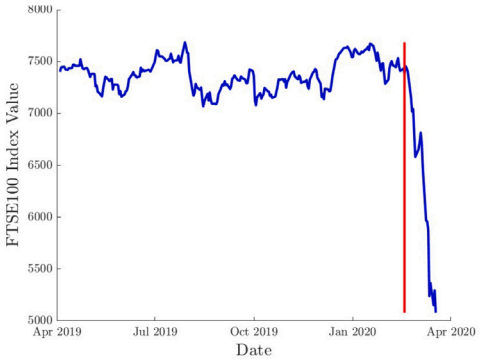
(a) Value of the CAC40 index. The red line indicates the beginning of the crisis.



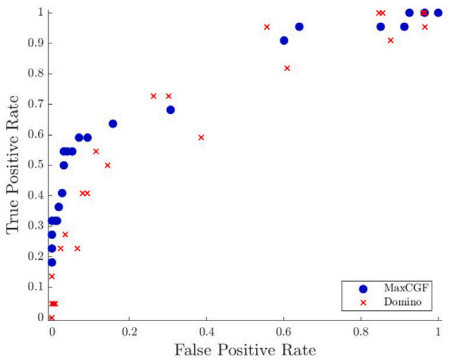
(b) ROC curves of the MaxCGF and Domino algorithms for the CAC40 dataset.

Fig. 12. CAC40 dataset.

]

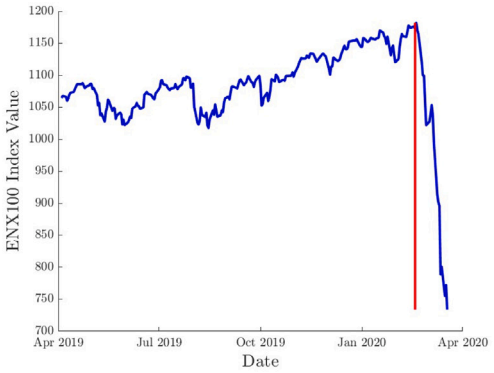


(a) Value of the FTSE100 index. The red line indicates the beginning of the crisis.

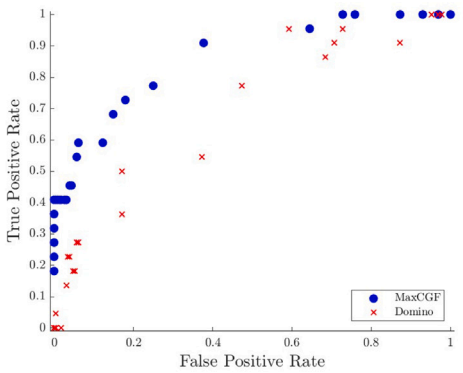


(b) ROC curves of the MaxCGF and Domino algorithms for the FTSE100 dataset.

Fig. 13. FTSE100 dataset.



(a) Value of the Euro Next 100 index. The red line indicates the beginning of the crisis.



(b) ROC curves of the MaxCGF and Domino algorithms for the N100 dataset.

Fig. 14. N100 dataset.

5. Conclusions

In this paper, we have proposed a non-parametric approach to detect anomalies in multivariate financial data by examining their univariate projections on appropriate directions that depend on cumulants of any order. Such directions, where the original data have been projected, are those that maximize the cumulant generating function (CGF).

In this respect, we have first refined some theoretical results of Bernacchia and Naveau [33], Bernacchia et al. [34] investigating the directions that maximize CGF of data with normal and skew-normal distributions. Then, we have proved in the general non-parametric case that CGF is a convex function and characterized the CGF maximization problem on the unit n-circle as a concave minimization problem. Furthermore, we have extended the outlier detection methodology based on the projections of multivariate data on the directions obtained by the classical PCA technique to the directions that maximize CGF. Finally, we have presented an extensive empirical analysis testing the performance of our outlier detection procedure, named MaxCGF, and comparing it with two other methods, proposed by Domino [32], Peña and Prieto [30]. From the computational results, we have observed that the Peña–Prieto method shows high performance for standard normal and normal simulated data, while, in other cases, its ability to detect outliers seems to be low. The Domino method typically provides intermediate performance, but favorable in the case of Student's t distributed data. The MaxCGF approach also performs well for normal and Student's random vectors. However, for skew-normal data or real-world financial data, the MaxCGF method always shows better results for all performance measures examined, including efficiency.

Ethical approval

This article does not contain any studies with human participants or animals performed by any of the authors.

Funding

This work was partially supported by the PRIN 2017 Project (no. 20177WC4KE), funded by the Italian Ministry of University and Research (MUR).

Appendix A. Relative variance of the CGF estimator

The sample estimate of the Cumulant Generating Function (CGF) of a discrete multivariate variable $\boldsymbol{X}_t = (X_{1,t}, \dots, X_{n,t})$ with $t = 1, \dots, T$ is

$$G_{\boldsymbol{X}}(\boldsymbol{\xi}) = \ln \frac{1}{T} \sum_{t=1}^{T} e^{\boldsymbol{\xi}^T \boldsymbol{X}_t} = \ln M_{\boldsymbol{X}}(\boldsymbol{\xi}),$$

where, therefore, $M_{\boldsymbol{X}}(\boldsymbol{\xi})$ denotes the sample estimate of the moment generating function. The relative variance of the CGF estimator is then defined as

$$\varepsilon_G^2 = \frac{\mathbb{V}\mathrm{ar}[G]}{(\mathbb{E}[G])^2}. \tag{31}$$

To find an explicit expression for (31), we exploit the Taylor expansion for moments of functions of random variables (see, e.g., [50]). More precisely, we use the Taylor expansion around $\mu_M = \mathbb{E}[M]$, namely

$$G = \ln M = \ln \mu_M + \frac{1}{\mu_M}(M - \mu_M) - \frac{1}{2}\frac{1}{\mu_M^2}(M - \mu_M)^2 + \cdots \tag{32}$$

Using the first order approximation, we obtain

$$\mathbb{E}[G] = \mathbb{E}[\ln M] \simeq \mathbb{E}[\ln \mu_M + \frac{1}{\mu_M}(M - \mu_M)] = \ln \mu_M \tag{33}$$

$$\mathbb{V}\text{ar}[G] = \mathbb{V}\text{ar}[\ln M] \simeq \mathbb{V}\text{ar}[\ln \mu_M + \frac{1}{\mu_M}(M - \mu_M)] = \frac{1}{\mu_M^2}\sigma_M^2 . \tag{34}$$

Thus,

$$\varepsilon_G^2 \simeq \frac{\sigma_M^2}{\mu_M^2(\ln \mu_M)^2} , \tag{35}$$

where $\mu_M = \mathbb{E}[M]$ and $\sigma_M^2 = \mathbb{V}\text{ar}[M]$.

The sample estimate of the moment generating function of a discrete multivariate variable $\boldsymbol{X}_t = (X_{1,t}, \dots, X_{n,t})$ with $t = 1, \dots, T$ is

$$M_{\boldsymbol{X}}(\boldsymbol{\xi}) = \frac{1}{T}\sum_{t=1}^{T} e^{\boldsymbol{\xi}^T \boldsymbol{X}_t} , \tag{36}$$

where, as mentioned in Section 2.4, to explicitly find (35), we assume i.i.d. normally distributed random vectors $\boldsymbol{X}_t \sim N_n(\boldsymbol{0}, \boldsymbol{\Sigma})$ $\forall t = 1, \dots, T$. Then, the expectation of the estimator (36) is

$$\mathbb{E}[M] = \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}[e^{\boldsymbol{\xi}^T \boldsymbol{X}_t}] , \tag{37}$$

where $\mathbb{E}[e^{\boldsymbol{\xi}^T \boldsymbol{X}_t}] = \int \cdots \int e^{\boldsymbol{\xi}\boldsymbol{x}} f_{\boldsymbol{X}_t}(\boldsymbol{x}) d\boldsymbol{x}$ and $f_{\boldsymbol{X}_t}(\boldsymbol{x}) = 2\pi^{-\frac{n}{2}} \det(\boldsymbol{\Sigma})^{-\frac{1}{2}} e^{-\frac{1}{2}\boldsymbol{x}^T \boldsymbol{\Sigma}^{-1}\boldsymbol{x}}$. Hence,

$$\begin{aligned}
\mathbb{E}[e^{\boldsymbol{\xi}^T \boldsymbol{X}_t}] &= 2\pi^{-\frac{n}{2}} \det(\boldsymbol{\Sigma})^{-\frac{1}{2}} \int \cdots \int e^{\boldsymbol{\xi}\boldsymbol{x}} e^{-\frac{1}{2}\boldsymbol{x}^T \boldsymbol{\Sigma}^{-1}\boldsymbol{x}} d\boldsymbol{x} \\
&= e^{\frac{1}{2}\boldsymbol{\xi}^T \boldsymbol{\Sigma}\boldsymbol{\xi}} 2\pi^{-\frac{n}{2}} \det(\boldsymbol{\Sigma})^{-\frac{1}{2}} \int \cdots \int e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\Sigma}\boldsymbol{\xi})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\Sigma}\boldsymbol{\xi})} d\boldsymbol{x} \\
&= e^{\frac{1}{2}\boldsymbol{\xi}^T \boldsymbol{\Sigma}\boldsymbol{\xi}} ,
\end{aligned} \tag{38}$$

since $2\pi^{-\frac{n}{2}} \det(\boldsymbol{\Sigma})^{-\frac{1}{2}} \int \cdots \int e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\Sigma}\boldsymbol{\xi})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\Sigma}\boldsymbol{\xi})} d\boldsymbol{x} = 1$. Thus, using (38) in (37), we obtain

$$\mu_M = \mathbb{E}[M] = \frac{1}{T}\, T\, e^{\frac{1}{2}\boldsymbol{\xi}^T \boldsymbol{\Sigma}\boldsymbol{\xi}} = e^{\frac{1}{2}\boldsymbol{\xi}^T \boldsymbol{\Sigma}\boldsymbol{\xi}} \tag{39}$$

For the variance of the estimator (36), we have

$$\mathbb{V}\text{ar}[M] = \mathbb{E}[M^2] - \mathbb{E}[M]^2 , \tag{40}$$

where

$$\begin{aligned}
\mathbb{E}[M^2] &= \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} e^{\boldsymbol{\xi}^T \boldsymbol{X}_t} \frac{1}{T}\sum_{j=1}^{T} e^{\boldsymbol{\xi}^T \boldsymbol{X}_j}\right] \\
&= \frac{1}{T^2}\mathbb{E}\left[\sum_{t=1}^{T} e^{2\boldsymbol{\xi}^T \boldsymbol{X}_t} + \sum_{t\neq j}^{T} (e^{\boldsymbol{\xi}^T \boldsymbol{X}_t})(e^{\boldsymbol{\xi}^T \boldsymbol{X}_j})\right] \\
&= \frac{1}{T^2}\sum_{t=1}^{T} \mathbb{E}\left[e^{2\boldsymbol{\xi}^T \boldsymbol{X}_t}\right] + \sum_{t\neq j}^{T} \mathbb{E}\left[(e^{\boldsymbol{\xi}^T \boldsymbol{X}_t})(e^{\boldsymbol{\xi}^T \boldsymbol{X}_j})\right] .
\end{aligned}$$

Now, similarly to (38), we obtain that

$$\mathbb{E}[e^{2\boldsymbol{\xi}\boldsymbol{X}_t}] = e^{2\boldsymbol{\xi}^T \boldsymbol{\Sigma}\boldsymbol{\xi}} . \tag{41}$$

Furthermore, since, by assumption, $\boldsymbol{X}_t$ are i.i.d. $\forall t = 1, \dots, T$, we have

$$\begin{aligned}
\mathbb{E}[M^2] &= \frac{1}{T^2}(T\mathbb{E}[e^{2\boldsymbol{\xi}^T \boldsymbol{X}}] + (T^2 - T)(\mathbb{E}[e^{\boldsymbol{\xi}^T \boldsymbol{X}}])^2) \\
&= \frac{1}{T}\mathbb{E}[e^{2\boldsymbol{\xi}^T \boldsymbol{X}}] + (1 - \frac{1}{T})(\mathbb{E}[e^{\boldsymbol{\xi}^T \boldsymbol{X}}])^2 \\
&= \frac{1}{T}e^{2\boldsymbol{\xi}^T \boldsymbol{\Sigma}\boldsymbol{\xi}} + (1 - \frac{1}{T})e^{\boldsymbol{\xi}^T \boldsymbol{\Sigma}\boldsymbol{\xi}} .
\end{aligned} \tag{42}$$

Substituting (42) in Expression (40), we can write

$$\begin{aligned}
\sigma_M^2 = \mathbb{V}\text{ar}[M] &= \frac{1}{T}e^{2\boldsymbol{\xi}^T \boldsymbol{\Sigma}\boldsymbol{\xi}} + (1 - \frac{1}{T})e^{\boldsymbol{\xi}^T \boldsymbol{\Sigma}\boldsymbol{\xi}} - e^{\boldsymbol{\xi}^T \boldsymbol{\Sigma}\boldsymbol{\xi}} \\
&= \frac{1}{T}\left(e^{2\boldsymbol{\xi}^T \boldsymbol{\Sigma}\boldsymbol{\xi}} - e^{\boldsymbol{\xi}^T \boldsymbol{\Sigma}\boldsymbol{\xi}}\right)
\end{aligned} \tag{43}$$

Thus, using (39) and (43) in (31),

$$\varepsilon_G^2 \simeq \frac{\sigma_M^2}{\mu_M^2(\ln \mu_M)^2} = \frac{\frac{1}{T}\left(e^{2\boldsymbol{\xi}^T \boldsymbol{\Sigma}\boldsymbol{\xi}} - e^{\boldsymbol{\xi}^T \boldsymbol{\Sigma}\boldsymbol{\xi}}\right)}{e^{\boldsymbol{\xi}^T \boldsymbol{\Sigma}\boldsymbol{\xi}}(\frac{1}{2}\boldsymbol{\xi}^T \boldsymbol{\Sigma}\boldsymbol{\xi})^2} = \frac{1}{T}\,\frac{e^{\boldsymbol{\xi}^T \boldsymbol{\Sigma}\boldsymbol{\xi}} - 1}{(\frac{1}{2}\boldsymbol{\xi}^T \boldsymbol{\Sigma}\boldsymbol{\xi})^2} . \tag{44}$$

Denoting $\xi = r\theta$, we obtain

$$\varepsilon_G^2 \simeq \frac{4}{T}\frac{e^{r^2\theta^T\Sigma\theta} - 1}{r^4(\theta^T\Sigma\theta)^2}\,. \tag{45}$$

Simplifying the issue, we substitute $\theta^T\Sigma\theta$ with its largest eigenvalue λ_1 (computed by the standard PCA), and, therefore, Expression (45) becomes

$$\varepsilon_G^2 \simeq \frac{4}{T}\frac{e^{r^2\lambda_1} - 1}{r^4(\lambda_1)^2}\,. \tag{46}$$

Appendix B. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.cam.2024.116457.

Data availability

Data will be made available on request.

References

[1] K. Singh, S. Upadhyaya, Outlier detection: applications and techniques, Int. J. Computer Sci. Issues (IJCSI) 9 (1) (2012) 307.
[2] F. Meng, G. Yuan, S. Lv, Z. Wang, S. Xia, An overview on trajectory outlier detection, Artif. Intell. Rev. 52 (4) (2019) 2437–2456.
[3] F. Maturo, F. Fortuna, T. Di Battista, Outliers detection in assessment tests' quality evaluation through the blended use of functional data analysis and item response theory, Ann. Oper. Res. 342 (2024) 1547–1562.
[4] T. Ané, L. Ureche-Rangau, J.-B. Gambet, J. Bouverot, Robust outlier detection for Asia–Pacific stock index returns, J. Int. Financial Mark. Inst. Money 18 (4) (2008) 326–343.
[5] I. Kondor, S. Pafka, G. Nagy, Noise sensitivity of portfolio selection under various risk measures, J. Bank. Financ. 31 (5) (2007) 1545–1573.
[6] F. Cesarone, F. Mango, C.D. Mottura, J.M. Ricci, F. Tardella, On the stability of portfolio selection models, J. Empir. Financ. 59 (2020) 210–234.
[7] R. Giacometti, G. Torri, S. Paterlini, Tail risks in large portfolio selection: penalized quantile and expectile minimum deviation models, Quant. Finance 21 (2) (2021) 243–261.
[8] T.S. Ferguson, On the rejection of outliers, in: Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Vol. 1, University of California Press Berkeley and Los Angeles, 1961, pp. 253–287.
[9] S.S. Wilks, Multivariate statistical outliers, Sankhyā: Indian J. Statist. Series A 40 (1963) 7–426.
[10] R. Gnanadesikan, J.R. Kettenring, Robust estimates, residuals, and outlier detection with multiresponse data, Biometrics (1972) 81–124.
[11] S.J. Schwager, B.H. Margolin, Detection of multivariate normal outliers, Ann. Statist. 10 (3) (1982) 943–954.
[12] R.D. Cook, Assessment of local influence, J. R. Stat. Soc. Ser. B Stat. Methodol. 48 (2) (1986) 133–155.
[13] L. Shi, X. Wang, Assessment of local influence in multivariate analysis, Acta Math. Sci. 16 (3) (1996) 257–270.
[14] L. Shi, Local influence in principal components analysis, Biometrika 84 (1) (1997) 175–186.
[15] R.D. Cook, S. Weisberg, Residuals and influence in regression, Chapman & Hall, 1982.
[16] I.S. Reed, X. Yu, Adaptive multiple-band CFAR detection of an optical pattern with unknown spectral distribution, IEEE Trans. Acoust. Speech Signal Process. 38 (10) (1990) 1760–1770.
[17] R. Das, B.K. Sinha, Detection of multivariate outliers with dispersion slippage in elliptically symmetric distributions, Ann. Statist. (1986) 1619–1624.
[18] B.K. Sinha, Detection of multivariate outliers in elliptically symmetric distributions, Ann. Statist. 155 (1984) 8–1565.
[19] R.A. Maronna, Robust M-estimators of multivariate location and scatter, Ann. Statist. 5 (1976) 1–67.
[20] W.A. Stahel, Breakdown of covariance estimators, 1981, Fachgruppe für Statistik, Eidgenössische Techn. Hochsch.
[21] D.L. Donoho, Breakdown properties of multivariate location estimators, Technical report, Harvard University, Boston, 1982.
[22] P.J. Rousseeuw, Least median of squares regression, J. Amer. Statist. Assoc. 79 (388) (1984) 871–880.
[23] P.J. Rousseeuw, Multivariate estimation with high breakdown point, Math. Statist. Appl. (1985).
[24] P.J. Rousseeuw, K. Van Driessen, A Fast Algorithm for the Minimum Covariance, Technometrics 41 (3) (1999) 212.
[25] M. Falk, On mad and comedians, Ann. Inst. Statist. Math. 49 (1997) 615–644.
[26] T.A. Sajesh, M.R. Srinivasan, Outlier detection for high dimensional data using the Comedian approach, J. Stat. Comput. Simul. 82 (5) (2012) 745–757.
[27] S. Shukla, S. Lalitha, Robust outlier detection method for multivariate spatial data, Nat. Acad. Sci. Lett. 44 (6) (2021) 551–554.
[28] D. Kazempour, M.A.X. Hünemörder, T. Seidl, On comads and principal component analysis, in: Similarity Search and Applications: 12th International Conference, SISAP 2019, Newark, NJ, USA, October (2019) 2–4, Proceedings 12, Springer, 2019, pp. 273–280.
[29] E. Cabana, R.E. Lillo, H. Laniado, Multivariate outlier detection based on a robust Mahalanobis distance with shrinkage estimators, Statist. Pap. 62 (2021) 1583–1609.
[30] D. Peña, F.J. Prieto, Multivariate outlier detection and robust covariance matrix estimation, Technometrics 43 (3) (2001) 286–310.
[31] D. Peña, F.J. Prieto, Combining random and specific directions for outlier detection and robust estimation in high-dimensional multivariate data, J. Comput. Graph. Statist. 16 (1) (2007) 228–254.
[32] K. Domino, Multivariate cumulants in outlier detection for financial data analysis, Phys. A (2020) 124995.
[33] A. Bernacchia, P. Naveau, Detecting spatial patterns with the cumulant function–Part 1: The theory, Nonlinear Process. Geophys. 15 (1) (2008) 159–167.
[34] A. Bernacchia, P. Naveau, M. Vrac, P. Yiou, Detecting spatial patterns with the cumulant function–Part 2: An application to El Niño, Nonlinear Process. Geophys. 15 (1) (2008) 169–177.
[35] A. Azzalini, A.D. Valle, The multivariate skew-normal distribution, Biometrika 83 (4) (1996) 715–726.
[36] R.B. Arellano-Valle, A. Azzalini, The centred parametrization for the multivariate skew-normal distribution, J. Multivariate Anal. 99 (7) (2008) 1362–1382.
[37] A. Azzalini, A. Capitanio, Statistical applications of the multivariate skew normal distribution, J. R. Stat. Soc. Ser. B Stat. Methodol. 61 (3) (1999) 579–602.
[38] E.T. Copson, E.T. Copson, Asymptotic expansions, (no. 55) Cambridge University Press, 2004.
[39] R.T. Rockafellar, Convex Analysis, vol. 36, Princeton University Press, 1970.
[40] H.P. Benson, Concave minimization: theory, applications and algorithms, in: Handbook of Global Optimization, Springer, 1995, pp. 43–148.
[41] P.M. Pardalos, J.B. Rosen, Methods for global concave minimization: A bibliographic survey, Siam Rev. 28 (3) (1986) 367–379.

[42] F.A. Al-Khayyal, R. Horst, P.M. Pardalos, Global optimization of concave functions subject to quadratic constraints: an application in nonlinear bilevel programming, Ann. Oper. Res. 34 (1992) 125–147.
[43] N.T. Trendafilov, I.T. Jolliffe, Projected gradient approach to the numerical solution of the scotlass, Comput. Statist. Data Anal. 50 (1) (2006) 242–253.
[44] D.E. Knuth, Art of Computer Programming: Seminumerical algorithms, vol. 2, Addison-Wesley Professional, 2014.
[45] J. Nocedal, S.J. Wright, Numerical optimization, Springer, 1999.
[46] P. Comon, Independent component analysis, a new concept? Signal Process. 36 (3) (1994) 287–314.
[47] A. Hyvärinen, E. Oja, Independent component analysis: algorithms and applications, Neural Netw. 13 (4–5) (2000) 411–430.
[48] LSEG, LSEG data & analytics, 2023.
[49] M. Shu, R. Song, W. Zhu, The COVID crash of the 2020 US Stock market, North Am. J. Econ. Finance 58 (2021) 101497.
[50] H. Benaroya, S.M. Han, M. Nagurka, Probability Models in Engineering and Science, vol. 192, CRC Press, 2005.