INFOSET: An R Package for Computing a New Informative Distribution Set for Asset Returns

Gloria Polinesi[1*], Maria Cristina recchioni[1†] and Francesca Mariani[1†]

[1*]Department of Economics and Social Sciences, Università Politecnica delle Marche, Piazzale Martelli 8, Ancona, 60121, Italy.

*Corresponding author(s). E-mail(s): g.polinesi@staff.univpm.it;
Contributing authors: m.c.recchioni@staff.univpm.it; f.mariani@staff.univpm.it;
†

Abstract

Hidden structure behind asset returns is a complex research area with relevant implications for modern portfolio theory and investment management. Going further in this direction, we introduce a dataset, hereafter informative set, which characterizes the left tail of financial series. The related INFOSET package computes this informative set via an adaptive clustering algorithm identifying sub-groups of gross returns at each iteration by approximating their distribution with a sequence of two-component log-normal mixtures. The package allows for assessing the reliability of the informative set while detecting common tail behaviors for clustering and predictive purposes. An endogenous tail risk measure is defined and tested for asset classification and portfolio construction. Finally, an illustration with ETF financial time series is provided to show its usage.

Keywords: Mixtures, tail risk, portfolio model, R

1 Introduction

In recent years dealing with unobserved patterns behind the distribution of financial returns has become a predominant topic in many research areas (see among others West (1993), Vlassis and Likas (1999) and MacDonald et al (2011)). Moreover, extreme events, despite their low probability, significantly contribute to market volatility (Gkillas and Longin (2018); Kapadia and Du (2011); Rossi and De Magistris (2013)). The 2008 financial crisis and COVID-19 recession have shown the limitations of traditional financial tools to manage these catastrophes (Agarwalla et al (2021); Gao et al (2019); Orlowski (2012)).

To address this issue, we propose a dataset, called the informative set, which allows for the identification of distinct dynamics in the left tail of time series data. We also introduce an endogenous risk metric, the Left Change Point Risk, denoted as LR_{cp}, defined by considering structural breakpoints in the left-hand tail of the time series distribution.

Specifically, the procedure to compute the informative set adjusts the method proposed by Mariani et al (2022a) to gross returns of financial assets. This is accomplished through an adaptive algorithm that identifies sub-groups of gross returns in each iteration by approximating their distribution with a sequence of two-component log-normal mixtures. These sub-groups emerge

when a significant change in the distribution occurs below the median of the financial returns, with their boundary termed as the "change point" of the mixture. The process concludes when no further change points are detected. The outcome encompasses parameters of the leftmost mixture distributions and change points of the analyzed financial time series.

The left change point risk, LR_{cp}, is a risk measurement based on the assumption that the existence of a change point below the median implies a structural breakpoint in the left-side of return distribution, so denoting the presence of turbulence and risk. Thus, the left change point risk, LR_{cp}, represents the expected loss below the change point threshold, making it endogenously defined.
This allows the association of binary or multilevel classification of the ETFs to be used in supervised machine learning techniques which are suitable for predicting financial time series data. Moreover, we propose to embed this risk into the Markowitz asset allocation following the work of Giudici et al (2022). Specifically, we modify the objective function of the Markowitz minimum variance portfolio taking into account not only the volatility of individual assets but also their left tail risk, expressed in terms of the LR_{cp}. Empirical results show that the resulting portfolios are less volatile and better performing than the classical Markowitz' portfolios.

The paper summarizes the adaptive procedure proposed by Mariani et al (2022a) and illustrates the INFOSET package in R for computing the left tail informative set of asset returns while illustrating properties and applications of the risk measure LR_{cp}. Specifically, we describe a 2-step procedure applied on asset returns through the main function infoset and its routine tail_mixture. Several useful packages have been developed in R to evaluate the risk of extreme market movements looking at the tails of the assets (see among others evir, qrmtools and extRemes packages). Differently from these tools, the INFOSET package is designed not to manage tail risk, but to identify characteristics of financial series by focusing on left tail behavior.

The informative set is then applied to asset classification via a standard clustering procedure mainly to test its ability to identify asset classes as in Mariani et al (2022b). The clustering approach deals with the specific features which describe the left tail distribution of financial returns. To the best of our knowledge, this use is an additional contribution of the informative set. In fact, the challenging task of clustering time series data tends to neglect these aspects. Far from being exhaustive, we cite the correlation-based clustering techniques of Mantegna (1999), Basalto et al (2007) and Giudici and Polinesi (2021). We also refer to Liao (2005) and Esling and Agon (2012) who provide a detailed discussion of time series clustering in the data mining literature.

The informative set deals with different purposes which are not limited to those we list here. Specifically, the functionalities of the INFOSET package include: (i) modelling asset distribution detecting the parameters which describe left tail behaviour, (ii) clustering, (iii) labeling of the financial series for predictive and classification purposes (iv) portfolio construction. Two datasets (sample.data and asset.label) are included in the R package INFOSET. The former involves daily observations of prices for the 44 ETFs considered. The latter includes the asset class of ETFs according to the classification provided by the Exchange where they are traded.

An outline of our paper is as follows. Section 2 provides a review of the informative set, while Section 3 illustrates how to compute it through the INFOSET R package. Section 4 presents an empirical application to ETFs time series which shows the effect of including the change point risk in well known portfolio optimization problems. Finally, Section 5 draws some conclusions.

2 Methodology

2.1 Left tail informative set

The gross return (or simply return) of the i-th asset, for $i = 1, \ldots, N_A$, at time t is defined as:

$$y_{t,i} = \exp\left\{\log\left(\frac{p_{t,i}}{p_{t-\Delta t,i}}\right)\right\} = \frac{p_{t,i}}{p_{t-\Delta t,i}},\ i = 1, 2, \ldots, N_A\,, \quad (1)$$

where $p_{t,i}$ is the daily price and Δt is the time step at which the prices are observed.

Under the assumption that the return distribution can be approximated by a bivariate

log-normal mixture, we apply the iterative stratification procedure introduced by Mariani et al (2022a). In the first iteration, the stratification procedure identifies the so-called change point and divides the gross returns into two distinct groups: returns that are smaller than or equal to the change point and returns that are larger than the change point. The change point is the threshold where the leftmost component of the mixture dominates the rightmost component to its left and is dominated by it to its right. This indicates that a structural change in the distribution occurs. In each subsequent iteration, the procedure approximates with a log-normal mixture the right group of returns from the previous iteration, appropriately shifted. It identifies the change point associated with this right group and, then, it splits this group into two sub-groups. The procedure stops when it fails to find a new change point or when the new change point is larger than the median of the returns.

Bearing in mind that the stratification procedure works similarly for each financial asset i, we drop the subscript i and, in the remainder of this section, we denote the gross return at time $t = t_j = j\Delta t$, with the simplified notation y_j, $j = 1, 2, \ldots, n$.

In its first iteration, the procedure identifies the first change point a^1 to split the set of all returns $\mathcal{S}_n = \{y_1, y_2, \ldots, y_n\}$ into two disjoint groups: the left group $\mathcal{K}_1 = \{y \in \mathcal{S}_n \wedge y \in (0, a^1]\}$, composed of returns smaller than or equal to the threshold value, and a right group $\mathcal{R}_1 = \mathcal{S}_n \backslash \mathcal{K}_1$, composed of returns larger than the threshold value.

In the second iteration, the procedure considers the subset $\mathcal{R}_1$, obtained in the first iteration. It identifies a new threshold value $a^2 > a^1$, and splits $\mathcal{R}_1$ into two disjoint groups: the left group $\mathcal{K}_2 = \{y \in \mathcal{S}_n \wedge y \in (a^1, a^2]\}$ and the right group $\mathcal{R}_2 = \{y \in \mathcal{S}_n \wedge y \in (a^2, +\infty)\}$. In the k-th iteration, the algorithm proceeds similarly to the first two iterations, by identifying the threshold a^k and dividing the set $\mathcal{R}_{k-1}$ into two groups: $\mathcal{K}_k = \{y \in \mathcal{S}_n \wedge y \in (a^{k-1}, a^k]\}$ and $\mathcal{R}_k = \{y \in \mathcal{S}_n \wedge y \in (a^k, +\infty)\}$.

The vector of unknown parameters for the density functions associated with the two mixture components $\underline{\Theta}_k = (\pi_k, \mu_{1,k}, \mu_{2,k}, \sigma_{1,k}, \sigma_{2,k})'$ is estimated using the return in the set $\mathcal{R}_{k-1}$ through the expectation maximization (EM) algorithm (see Dempster et al (1977)). Note that $\pi_k \in [0, 1]$ is the mixing weight representing the a priori probability that the point $x = y - a^{k-1}$, $y \in \mathcal{K}_{k-1}$, for $k = 1, 2, ...$, belongs to the first component. Hereafter we assume $a^0 = 0$.

The change point a^k of the mixture is determined using the following rule:

$$\begin{aligned} a^k = \min\{y \in \mathcal{R}_{k-1} \ \wedge \ & \pi_k f_{1,k}(y - a^{k-1}) \\ & = (1 - \pi_k) f_{2,k}(y - a^{k-1})\}. \end{aligned} \tag{2}$$

where $f_{1,k}(x)$ and $f_{2,k}(x)$, $x \in \mathbb{R}_+$, are the log-normal densities of parameters $\mu_{1,k}$, $\mu_{2,k}$, $\sigma_{1,k}$, $\sigma_{2,k} \in \mathbb{R}$ associated with the two mixture components.

An explicit formula for the change points is

$$a^k = \exp\left\{\min\{\log(a_+^k), \log(a_-^k)\}\right\}, \tag{3}$$

where $\log(a_+^k)$, $\log(a_-^k)$ are given by

$$\log(a_\pm^k) = \frac{\sigma_{1,k}^2 \sigma_{2,k}^2}{\sigma_{2,k}^2 - \sigma_{1,k}^2} \left[\left(\frac{\mu_{1,k}}{\sigma_{1,k}^2} - \frac{\mu_{2,k}}{\sigma_{2,k}^2} \right) \pm \sqrt{\Delta_k} \right], \tag{4}$$

with Δ_k defined as

$$\begin{aligned} \Delta_k = & \left(\frac{\mu_{1,k}}{\sigma_{1,k}^2} - \frac{\mu_{2,k}}{\sigma_{2,k}^2} \right)^2 + \left(2 \log \left(\frac{\sigma_{2,k} \pi_k}{\sigma_{1,k}(1 - \pi_k)} \right) \right. \\ & \left. - \left(\frac{\mu_{1,k}}{\sigma_{1,k}} \right)^2 + \left(\frac{\mu_{2,k}}{\sigma_{2,k}} \right)^2 \right) \frac{\sigma_{2,k}^2 - \sigma_{1,k}^2}{\sigma_{1,k}^2 \sigma_{2,k}^2}. \end{aligned} \tag{5}$$

The change point a^k is the frontier of the two groups $\mathcal{K}_k$ and $\mathcal{R}_k$ at the k-th iteration and, broadly speaking, a^k divides the sample into two subsamples with non-homogeneous distributions. The procedure stops when a new a^k cannot be determined (i.e., Eq. (2) does not admit any solution) or the change point is larger than the median of returns.

The informative set for the left tail is identified by applying the stratification procedure up to the second iteration.

At time t, for $i = 1, 2, \ldots, N_A$ each asset i is identified by the following twelve parameters:

1. The parameters (drift and volatility) of the left-hand component of the log-normal mixture: $\sigma^1_{1,t,i}$, $\mu^1_{1,t,i}$, $\sigma^2_{1,t,i}$, $\mu^2_{1,t,i}$.
2. The change points of the asset return distribution: $a^1_{t,i}$ and $a^2_{t,i}$.
3. The complement of the cumulative distribution functions, $1 - F^1_{1,t,i}$, $1 - F^2_{1,t,i}$, associated with the leftmost component of the mixture (first type error).
4. The cumulative distribution functions associated with the rightmost component of the mixture, $F^1_{2,t,i}$, $F^2_{2,t,i}$ (second type error).
5. The a priori probabilities: $\pi^1_{t,i}$, $\pi^2_{t,i}$.

The twelve parameters of each asset are collected into the matrix $X_t \in R^{N_A \times 12}$, the left tail informative set at time t. For simplicity, we consider the information coming from the first two iterations (i.e., $k = 1, 2$).

It is worth noting that the informative set can be computed when two main assumptions are satisfied: a) the gross return distribution can be approximated with a non-trivial two-component log-normal mixture with a change point; and b) the expectation maximization algorithm converges. The first assumption can be tested using, for example, the results in Chen and Li (2009) and Chauveau et al (2019). The assumptions for the EM approach to converge are no time interval with constant prices, no extreme outliers, and a sufficiently large sample size, as detailed in Yang and Chen (1998).

In the context of mixture distribution, the EM algorithm works in the usual way, maximizing the log-likelihood function associated with the observation vector $\underline{y} = (y_1\, y_2\, \ldots, y_n)'$ via a two-step procedure. This procedure initializes the unknown mixture parameters π, μ_1, μ_2, σ_1, σ_2, and, at any step, first evaluates the posterior probabilities (E-step) and next estimates the new mixture parameters $\hat{\pi}$, $\hat{\mu}_1$, $\hat{\mu}_2$, $\hat{\sigma}_1$, $\hat{\sigma}_2$ (M-step).

2.2 Left tail risk measure and portfolio strategies

In this section we compute the LR_{cp} risk measure[1] based on the first change point. Specifically, it aims to estimate the expected loss given that the loss has not exceeded the first change point threshold. Therefore, LR_{cp} can be represented as:

$$\begin{aligned} LR_{cp} &= \mathbb{E}[-X \mid X \leq cp] \\ &= -\frac{1}{F(cp)} \int_{-\infty}^{cp} x f(x)\, dx, \end{aligned} \tag{6}$$

where X and cp are, respectively, the logarithm of the gross return and the logarithm of the change point, while f is the probability density function (PDF) of X and $F(cp)$ represents the cumulative distribution function (CDF) of X evaluated at cp.

Note that the left change point risk, LR_cp, coincides with the Tail Value at Risk (TVaR) at level α, where $\alpha = F(cp)$. Unlike TVaR, where the level is predetermined, in this case, the level is found endogenously through the stratification procedure.

Moreover, the informative set can potentially be used to compose portfolios with specific degrees of risk. Specifically, we solve the following asset allocation problem:

$$\begin{aligned} &\min_{\underline{w}} \underline{w}' COV\, \underline{w} + \lambda \sum_{i=1}^{N_A} LR^i_{cp} w_i, \\ &\sum_{i=1}^{N_A} w_i x_i = \overline{x}, \quad \sum_{i=1}^{N_A} w_i = 1, \quad w_i \geq 0, \end{aligned} \tag{7}$$

where $x_i = log(y_i)$ is the logarithm of the gross return associated with the $i-$th ETF for $i = 1, 2, \ldots, N_A$ and N_A is the number of ETFs, $\overline{x}$ is chosen to be the average of the returns over the time period considered and LR^i_{cp} is the left risk associated with the ETF i. The coefficient λ allows investors to choose a risk profile by suitably weighting the two risks: asset covariance risk and left change point risk.

We compare four different asset allocation strategies. The first two are the classical and combined Markowitz selections, obtained by setting the matrix COV in Eq. (7) to the sample variance-covariance matrix with $\lambda = 0$ and $\lambda = 10^{-4}$, respectively.[2] The last two are the classical and combined extreme downside correlation (EDC)

[1] It satisfies the following properties: (i) weakly monotonicity, (ii) positive homogeneity and (iii) translation invariance. For more details about properties for measures of risk we refer to Artzner et al (1999).

[2] The value $\lambda = 10^{-4}$ ensures that the two risks are comparable in magnitude.

selections, obtained by setting COV in Eq. (7) to the extreme downside covariance matrix, as defined in Harris et al (2019) and Ahelegbey et al (2021), with $\lambda = 0$ and $\lambda = 10^{-4}$, respectively.

The portfolio analysis is conducted on consecutive overlapping time windows of FT consecutive trading days, with two consecutive windows differing by ov consecutive trading days, for a total of tw time windows. We use $\tau_{j-1} = 1 + ov(j-1)$ and $\tau_j = FT + ov(j-1)$ to denote the first and last observation times of the j-th window, $j = 1, 2, \ldots, tw$.

For any strategy, we solve tw different allocation problems in the form (7) using the asset returns observed in the above-mentioned time windows, and $w_{i,j}$ represents the weight of the i-th asset in the portfolio, $i = 1, 2, \ldots, N_A$, computed using the asset returns observed over the j-th window.

We compare the portfolio variability measures looking at the portfolio return up to ov days ahead. Specifically, for $j = 1, 2, \ldots, tw$, we define the portfolio return as

$$r_{t,j} = \frac{\left(\sum_{i=1}^{N_A} w_{i,j} p_{t,i}\right) - p_j}{p_j}, \quad t = \tau_j + 1, \tau_j + 2, \ldots, \tau_j + ov, \tag{8}$$

where p_j is the value of the portfolio on the last date of the j-th window.

3 The infoset function and its routine

In the R package INFOSET, the leftmost informative set $X_t \in R^{N_A \times 12}$ can be obtained by applying the 2-step stratification procedure on the gross returns as defined in Eq. (1). In the first step, the function infoset calls the routine tail_mixture, which computes the the log-normal mixture applied on the gross returns (y) and identifies the first change point a^1 solving Eq. (2). In the second step, the original gross returns are shifted by the value of change point a^1 and the change point , a^2, of the log-normal mixture associated with the shifted gross returns $(y - a^1)$ is identified. The change points together with the mean, the volatility, the a priori probabilities, the first and second type errors of the log-normal mixtures found at the first and second iterations are collected in the informative set.

The function tail_mixture is the routine for the infoset function and it takes the following arguments:

- y: a data.frame or matrix containing the observations at time t for the N_A time series;
- shift: the starting change point which is initialized to zero.
- n_it: the number of iterations.

The function tail_mixture considers the values y larger than the starting change point shift (i.e. the threshold value mentioned above) and shifts these values in a way such that the shifted values range in the interval $(0, +\infty)$. Then, starting from the shifted values, the function computes through the EM algorithm the corresponding log-normal mixture and the corresponding change point according to Eq. (3). The function returns a list object with a class attribute "tail_mixture" with ten elements: the change point, a flag variable, the mean and the standard deviation of the two mixture components, the value of the log-normal mixture components in the change point and a plot. A flag variable equal to zero indicates success (i.e., Eq. (2) admits solution), a flag variable equal to one indicates failure (i.e., Eq. (2) does not admit any solution).

```
tail_mixture <- function(y, shift, n_it){
     vec = which(y > shift)
     y = y[vec] - shift
     set.seed(600)
     GMModel = normalmixEM(log(y),
     lambda = NULL,
     mu = NULL, sigma = NULL, k = 2,
     mean.constr = NULL,
     sd.constr = NULL, epsilon =
     1e-08, maxit = 1000,
     maxrestarts = 500,
     verb = FALSE, fast = FALSE,
     ECM = FALSE, arbmean
     = TRUE, arbvar = TRUE)
     zsol = numeric()
     zsol[1] = GMModel$mu[1]
     zsol[3] = GMModel$mu[2]
     zsol[2] = GMModel$sigma[1]
     zsol[4] = GMModel$sigma[2]
     pa = GMModel$lambda[1]
```

```
 h1 = pa*dlnorm(min(log(y)),
 zsol[1], zsol[2])
 h2 = (1 - pa)*dlnorm(min(log(y)),
 zsol[3], zsol[4])
 indec = as.numeric(which.max(c(h1,
 h2)))
 if(indec == 2){
        mu1 = zsol[3]
        sigma1 = zsol[4]
        ppa = (1 - pa)
        mu2 = zsol[1]
        sigma2 = zsol[2]
}
else{
        mu1 = zsol[1]
        sigma1 = zsol[2]
        ppa = pa
        mu2 = zsol[3]
        sigma2 = zsol[4]
}
delta=(mu2*sigma1^2 - mu1*sigma2^2)^2
+ (mu2^2*sigma1^2 -
mu1^2*sigma2^2 +
2*sigma1^2*sigma2^2*
log(sigma2*ppa/(sigma1*(1 - pa))))*
(sigma2^2 - sigma1^2)
if(delta <= 0){
         if(n_it==1){
                flag = 1
                a = 0
                inters = 1
                inters1 = 1
                ppa = 1
                stop('no
                subpopulations',
                '\ n')
              }
         else{
              flag = 1
              a = 0
              inters = 1
              inters1 = 1
              ppa = 1
              stop('only one
              subpopulation
              exists', '\n')
            }
}
else{
    aa = min(exp((-(mu2*sigma1^2
    - mu1*sigma2^2) -
        sqrt(delta))/(sigma2^2
        - sigma1^2)),
        exp((-(mu2*sigma1^2 -
        mu1*sigma2^2)
        sqrt(delta))/(sigma2^2 -
        sigma1^2)))
    }
    flag = 0
    inters = plnorm(aa, mu2, sigma2)
    inters1 = (1-plnorm(aa, mu1,
    sigma1))
    a = aa + shift
    x1 = seq(from = min(y), to = max(y),
    by = (max(y) -  min(y))/1000)
    mix1 = dlnorm(x1, mu1, sigma1)
    mix2 = dlnorm(x1, mu2, sigma2)
    kern <- density(y, n = 1001)
    f = kern$y
    ff1 = ppa*mix1 + (1 - ppa)*mix2
    plot = plot(x1, f)
    hist(y, freq = FALSE)
    lines(x1, ff1, col = 'red')
    lines(x1, mix1*ppa)
    lines(x1, mix2*(1 - ppa))
    points(aa, ppa*dlnorm(aa,
    mu1, sigma1),
    col = red')
    Sys.sleep(5)
    return (c(a, flag, mu1,
    sigma1, mu2,
    sigma2, ppa,
    inters, inters1, plot))
}
```

The function infoset is the main function and it takes in input y, which is a data.frame or a matrix object containing the observations at time t for the N_A time series. It computes the left tail informative set at time t associated with the input data y. Specifically, the function infoset performs the two iterations of the stratification procedure, explained above. The function initializes the change point to zero and, starting from the input data y, it calls the routine tail_mixture to determine the first change point a^1 and the parameters of the log-normal mixture associated with y. This is the first iteration of the stratification procedure. Then, it calls again the routine tail_mixture using as input y and the first change point a^1 to determine the second change point a^2 and the parameters of the log-normal mixture associated with the values of y larger than

the first change point (i.e., the right group mentioned above). Finally, the elements of the left tail informative set are collected in a list.

We highlight that the aforementioned functions have a few exceptions associated with specific situations:

(i) Eq. (2) does not have solutions at all when the gross returns are drawn from a log-normal distribution.

(ii) Only the first iteration works. The latter can result from two different scenarios:

ii.a) Only one subpopulation exists because $\Delta <= 0$ for the second iteration.

ii.b) The first change point is larger than or equal to the median value (an unacceptable solution), indicating that the time series is too short.

```
infoset <- function(y){
    shift = NULL
    shift[1] = 0
    k = 1
    flag = 0
    meant = NULL
    dev = NULL
    meanr = NULL
    devr = NULL
    pa = NULL
    cum = NULL
    cum1 = NULL
    out = NULL
    shift = NULL
    shiftt = 0
    while(flag == 0 & shiftt < median(y)
        & k<3){
            out <- tail_mixture(y,
            shiftt, k)
            k = k+1
            shiftt = out[1]
            if(shiftt>=median(y)){
            print('Not valid
            change point:
            time series too short')
            }
            shift[(k-1)] <- out[1]
            flag <- out[2]
            meant[(k-1)] <- out[3]
            dev[(k-1)] <- out[4]
            meanr[(k-1)] <- out[5]
            devr[(k-1)] <- out[6]
            pa[(k-1)] <- out[7]
            cum[(k-1)] <- out[8]
            cum1[(k-1)] <- out[9]
     }
    if(flag == 1){
        shift = shift[1:(length(shift)
        - 1)]
        cum = cum[1:(length(cum)
        - 1)]
        cum1 = cum1[1:(length(cum1)
        - 1)]
        meant = meant[1:(length(meant)
        - 1)]
        dev = dev[1:(length(dev)
        - 1)]
        pa = pa[1:(length(pa)
        - 1)]
    }
    list('change point' = shift,
    'prior probability' = pa,
    'first type error' = cum,
    'second type error' = cum1,
    'mean' = meant, 'sd' = dev)
}
```

4 Example

To illustrate how the R package INFOSET works in practical situations, we present an empirical application with multivariate time series of ETFs. Starting from daily prices of $N_A = 44$ ETFs, available in the dataset sample.data, we define the informative set which includes, among others, the parameters of the leftmost mixture distributions and the change points of all the asset time series analyzed. The informative set is then applied to asset classification via a standard clustering procedure mainly to test its ability to identify asset classes (included in the dataset asset.label). Next, it is used to compute a risk measure based on the change points for labeling assets, which is particularly useful when their classification is not readily available, as well as for improving the performance of well-known portfolio optimization formulation.

The dataset used to compute the informative set includes the closing prices of ETFs from January 2006 to February 2018, for a total of 3,174 observations per series. The dataset can be loaded in the R workspace using:

```
data("sample.data", package = "INFOSET")
```

where sample.data is a 3,174 $\times$ 44 data.frame object containing the daily prices. As a first step,

we calculate the $N_A - 1$ daily series of gross returns through the g_ret function:

```
gross.ret<-as.data.frame(lapply(
           sample.data, g_ret))
```

where gross.ret is a 3,173 × 44 data.frame object containing the daily returns. Second, we estimate the mostleft information set for each ETF by using the following code[3]:

```
result <- NULL
for(i in 1:ncol(gross.ret)){
result[[i]] <- infoset(gross.ret[,i],
               plot_cp = "F")
}
output <- matrix(unlist(result), 12,
          ncol = ncol(gross.ret))
output <- t(output)
rownames(output) <- colnames(gross.ret)
colnames(output) <- c('ch_1', 'ch_2',
                'priori_1', 'priori_2',
                'first_1', 'first_2',
                'second_1', 'second_2',
                'mean_1', 'mean_2',
                'dev_1', 'dev_2')
output <- as.data.frame(output)
```

where output is a 44 × 12 data.frame object containing the estimated parameters.

```
head(round(output,4))
        ch_1   ch_2 priori_1
ETF_1 0.9972 0.9985   0.2313
ETF_2 0.9943 0.9968   0.8068
ETF_3 0.9986 0.9992   0.3943
ETF_4 0.9943 0.9967   0.8068
ETF_5 0.9851 0.9912   0.7411
ETF_6 0.9831 0.9900   0.7875
       priori_2 first_1 first_2
ETF_1 0.8289  0.0145  0.5777
ETF_2 0.1381  0.1443  0.0084
ETF_3 0.1947  0.0538  0.0163
ETF_4 0.8674  0.1439  0.5630
ETF_5 0.1535  0.1938  0.0095
ETF_6 0.1435  0.2126  0.0078
       second_1 second_2 mean_1
ETF_1 0.8119   0.9884 -1e-04
ETF_2 0.9910   0.4359  1e-04
ETF_3 0.7462   0.4237 -1e-04
ETF_4 0.9910   0.9924  1e-04
ETF_5 0.9817   0.3952  3e-04
ETF_6 0.9875   0.4319  4e-04

       mean_2  dev_1  dev_2
ETF_1 -5.7152 0.0031 0.3918
ETF_2 -6.1732 0.0025 1.0537
ETF_3 -7.5187 0.0020 1.1611
ETF_4 -5.0884 0.0025 0.3850
ETF_5 -5.4050 0.0073 1.1515
ETF_6 -5.1838 0.0078 1.2373
```

If the optional argument plot_cp is set to TRUE, a plot of the kernel density and a plot comparing the empirical distribution with the underlying log-normal mixture (and the change point) are generated for each gross return computed according to Eq. (1) (see, Fig. 1).

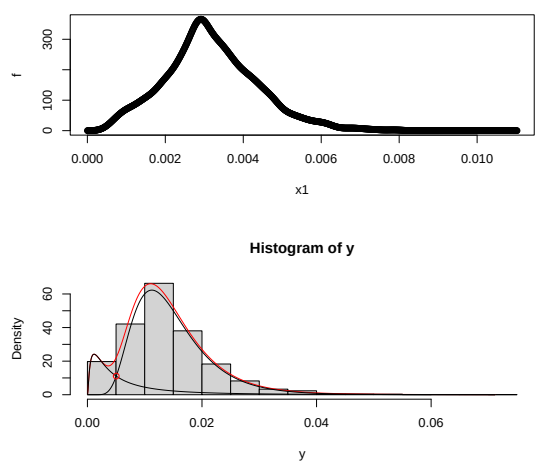

Fig. 1 Plots of results from infoset estimation for a single time series of gross returns. The top panel shows the kernel density of the time series. The bottom panel shows the empirical distribution and its underlying log-normal mixture highlighting the change point (red point).

4.1 Potential usages of INFOSET

Potential usages of the data object output involve clustering, labeling and portfolio modeling. In the following subsections, we provide details about these three applications available in the list of functions associated with the INFOSET package.

[3] After the execution of the code, a counter with the number of the iteration in the estimation function is shown until convergence or the maximum number of iterations is reached.

4.2 Clustering

According to the clustering purpose, Fig. 2 shows the dendrogram associated with the complete linkage algorithm applied to the Euclidean distance computed from the informative set:

$$\overline{d}_{t,i,j} = (\underline{x}_{t,i} - \underline{x}_{t,j})'(\underline{x}_{t,i} - \underline{x}_{t,j}), \quad i,j = 1,2,\ldots,N_A, \tag{9}$$

where

$$\underline{x}_{t,i} = (a^1_{t,i}, a^2_{t,i}, \pi^1_{t,i}, \pi^2_{t,i}, 1 - F^1_{1,t,i}, F^2_{1,t,i}, F^1_{2,t,i}, F^2_{2,t,i}, \mu^1_{1,t,i}, \mu^2_{1,t,i}, \sigma^1_{1,t,i}, \sigma^2_{1,t,i})'$$

and, analogously,

$$\underline{x}_{t,j} = (a^1_{t,j}, a^2_{t,j}, \pi^1_{t,j}, \pi^2_{t,j}, 1 - F^1_{1,t,j}, F^2_{1,t,j}, F^1_{2,t,j}, F^2_{2,t,j}, \mu^1_{1,t,j}, \mu^2_{1,t,j}, \sigma^1_{1,t,j}, \sigma^2_{1,t,j})'$$

.

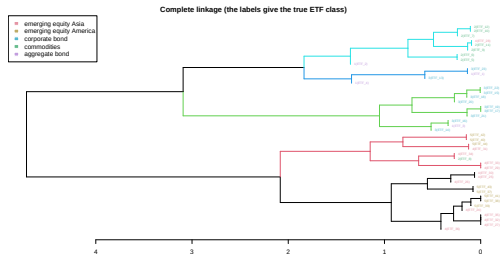

Fig. 2 Dendrogram associated with complete linkage clustering based on distance (9).

Fig. 2 shows the result obtained by the clustering procedure with dissimilarity matrix (9), which can be reproduced by running the following instructions:

```
library(dendextend)
library(colorspace)

data('asset.label', package = 'INFOSET')
group_label <- as.factor(
asset.label$label)
d <- dist(output, method = 'euclidean')
hc_SIMS <- hclust(d, method = 'complete')
dend_SIMS <- as.dendrogram(hc_SIMS)
dend_SIMS <- color_branches(dend_SIMS,
             k = 4, col = c(1:4))
labels_colors(dend_SIMS) <-
rainbow_hcl(5)
[sort_levels_values(
as.numeric(group_label)
[order.dendrogram(
dend_SIMS)])]
labels(dend_SIMS) <-
paste(as.character
(group_label)[order.dendrogram(
dend_SIMS)],
'(', labels(dend_SIMS), ')',
sep = '')
dend_SIMS <- hang.dendrogram(
dend_SIMS,
hang_height = 0.001)
dend_SIMS <-
assign_values_to_leaves_nodePar
(dend_SIMS, 0.5, 'lab.cex')
dev.new()
par(mar = c(1.8, 1.8, 1.8, 1))
plot(dend_SIMS, main = 'Complete linkage
(the labels give the true ETF class)',
horiz = TRUE, nodePar = list(cex
= 0.007))
labels_colors(dend_SIMS)
legend('topleft', legend = c('emerging
equity Asia',
'emerging equity
America', 'corporate bond',
'commodities',
'aggregate bond'), fill = c('#E495A5',
  '#BDAB66',
  '#55B8D0', '#65BC8C',
  '#C29DDE'),
  border = 'white')
```

Despite its sensitivity to outliers, complete linkage clustering avoids the chaining effect suffered by single linkage clustering, thereby, it is usually preferred to single linkage. Table 1 presents the findings related to the analysis's accuracy. To streamline the data, the emerging classes (Asia and America) were combined into a single category labeled as "emerging".

4.3 Classification and asset allocation

In this example, we use a rolling time windows of length $FT = 1290$ consecutive days (approximately 5 trading years) and two consecutive

Table 1 Classification resulting from the complete linkage clustering based on distance (9) (rows and columns represent the true asset classes and the predicted ones by cluster groups, respectively).

Actual/Predicted	aggr.	comm.	corp.	em.
aggr.	2	1	1	0
comm.	0	7	0	1
corp.	2	0	9	0
em.	0	1	0	20

windows differ by $ov = 125$ (approximately 6 trading months)[4].

Specifically, to classify assets and then to incorporate the measure LR_{cp} into the portfolio model, we compute its values over the time windows. This approach enables a dynamic perspective on risk assessment, as LR_{cp} values are updated and adapted to evolving market conditions. The calculated LR_{cp} values across these windows are stored for each asset, forming a panel data frame that is essential for further portfolio analysis and risk classification.

Here, LR_{cp} values are computed for each asset over each specified time window, producing a list object, LR, of length tw. This structure is instrumental in tracking changes in LR_{cp} over time, thus facilitating time-sensitive asset classification and risk management.

```
LR <- LR_cp(sample.data, FT= 1290,
      ov = 125)
df <- as.data.frame(matrix(unlist(LR),
      nrow = length(LR),
      ncol = ncol(sample.data),
      byrow = T))
df<-t(df)
colnames(df) <- c(paste("tw",
rep(1:16)))
head(df)
```

```
         tw 1        tw 2        tw 3
1 0.004115554 0.004122932 0.004134215
2 0.008701516 0.008763616 0.008753123
3 0.002539979 0.002580082 0.002596818
4 0.008711204 0.008772693 0.008762419
5 0.022961792 0.023126742 0.023292736
6 0.026571584 0.027158223 0.027269359
        tw 4        tw 5        tw 6
1 0.004173533 0.004228381 0.004220162
2 0.008704115 0.008723058 0.008777091
3 0.002625040 0.002608169 0.002529951
4 0.008714391 0.008732963 0.008773772
5 0.023250445 0.023270412 0.023008188
6 0.027266553 0.027212254 0.027192315
        tw 7        tw 8        tw 9
1 0.004166170 0.003986231 0.003963061
2 0.008834624 0.008103924 0.007892905
3 0.002428290 0.002409038 0.002350246
4 0.008840285 0.008111546 0.007886221
5 0.021672476 0.021033909 0.021293727
6 0.025415436 0.023289728 0.023276834
       tw 10       tw 11       tw 12
1 0.003990029 0.004260559 0.004337454
2 0.007707857 0.007836256 0.007724300
3 0.002393488 0.002481345 0.002472332
4 0.007712620 0.007834515 0.007769396
5 0.021741879 0.021031284 0.020499805
6 0.023778444 0.023871416 0.023351993
        tw 13       tw 14       tw 15
1  0.004358483 0.004340635 0.004322432
2  0.008290197 0.008509243 0.008478295
3  0.002445175 0.002487457 0.002488479
4  0.008308656 0.008617359 0.008582162
5  0.020145876 0.019997366 0.020072072
6  0.022291894 0.022133957 0.022091486
        tw 16
1 0.004278957
2 0.008459254
3 0.002484978
4 0.008560500
5 0.020057752
6 0.022023856
```

Figure 3 shows the LR_{cp} (y-axis) for the first time window as a function of the ETF index labeled from 1 to 44 (x-axis), with different shapes corresponding to the asset class: filled circle (aggregate bond), square (commodities), triangle (corporate bond), star (emerging equity Asia), and hollow circle (emerging equity America).

```
plot(df[,1], pch=19,
     col=asset.label$label,
     ylab = 'LR_cp', xlab = 'ETFs')
```

The significance of LR_{cp} in the portfolio model is underscored by its utility in labeling assets based on empirical risk levels. This is particularly valuable when asset classifications are either unavailable or insufficient for risk modeling. For example, Fig. 3 suggests three levels of risks (low, middle and high) according to the values of LR_{cp}: $(0;\ 0.01)$, $(0.01;\ 0.03)$ and $(0.03;\ +\infty)$.

[4]Note that, the function create_overlapping_windows, called by the functions LR_cp and ptf_construction and also available for own use, allows user to analyze different periods by setting values for the arguments FT and ov.

These categories are consistent with typical investment risk profiles, supporting the use of LR_{cp} in supervised machine learning techniques, such as classification or clustering, for predictive modeling in portfolio optimization. The rolling-window calculation of LR_{cp} thus not only enhances asset labeling precision but also provides a foundation for adaptive risk management strategies in dynamic market environments.

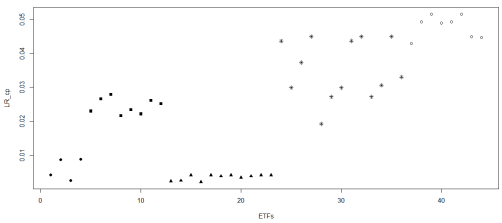

Fig. 3 LR_{cp} as a function of ETF index. Shapes correspond to the asset class: filled circle (aggregate bond), square (commodities), triangle (corporate bond), star (emerging equity Asia) and hollow circle (emerging equity America).

Once the LR_{cp} are computed, they can be included into portfolio model. In the R package INFOSET, the asset allocation strategy according to Eq. (7) can be performed by using the function ptf_construction, which takes the following arguments.

- Data: a (t x N_A) matrix or data.frame containing the N_A time series over period t.
- FT: window length.
- ov: number of different days for two consecutive time windows.
- ptf: type of portfolio to be computed: "M" is the Markowitz portfolio, "C_M" is the combined Markowitz portfolio, "EDC" uses the extreme downside correlation and "C_EDC" is the combined extreme downside correlation portfolio.
- LR_cp_measure: object of class LR_cp (only for "C_M' and "C_EDC" asset allocation strategies).

Code to implement classical and combined Markowitz strategy[5] is as follows:

```
ptf_results_M <-
ptf_construction(sample.data,
    FT = 1290, ov = 125, ptf = 'M')
ptf_results_C_M <-
    ptf_construction(sample.data,
    FT = 1290, ov = 125,
LR_cp_measure = LR, ptf = 'C_M')
```

The ptf_construction function returns two lists: the weights (ptf.weights) of assets in the portfolio by solving Eq. (7), along with the out of sample returns (ptf.oos.values) according to Eq. (8), for all the time windows considered in the analysis.

Here, the results for the first time window are provided:

```
round(ptf_results_M$ptf.weights[[1]], 3)

 [1] 0.000 0.000 0.000 0.134
 [5] 0.000 0.000 0.000 0.000 0.000
[10] 0.000 0.000 0.000 0.132 0.000
[15] 0.000 0.000 0.000 0.222 0.000
[20] 0.000 0.222 0.000 0.000 0.000
[25] 0.000 0.000 0.000 0.225 0.000
[30] 0.000 0.000 0.000 0.000 0.000
[35] 0.000 0.000 0.066 0.000 0.000
[40] 0.000 0.000 0.000 0.000 0.000

head(round(
    ptf_results_M$ptf.oos.value[[1]],
    3))

     1289   1290   1291   1292
[1,]    0 -0.003 -0.019 -0.023
     1293   1294   1295   1296
-0.017 -0.006 -0.011 -0.023
    1297      1298   1299   1300
 -0.031      -0.022 -0.032 -0.018
```

Figure 4 shows summary plots obtained by using the available plot_ptf function for aggregated portfolio returns which automatically include the median (black line) and mean (red line) values.

```
plot_ptf(ptf.oos.values)
```

It is also possible to summarize the output of the ptf_construction using the summary_ptf function:

```
summary_ptf(ptf.oos.values =
ptf_results_M$ptf.oos.value)

     Min.   1st Qu.    Median
-0.108177 -0.026327 -0.001541
     Mean   3rd Qu.      Max.
-0.002449  0.015506  0.112578
```

[5] See help("ptf_construction") for details about specifying starting values for this function.

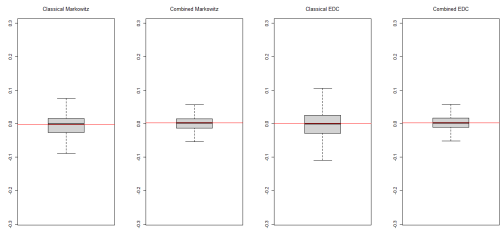

Fig. 4 Summary plots for aggregated portfolio returns.

Note that the combined portfolios show better performance in terms of higher profits and less volatility with respect to the portfolios obtained by applying the Markowitz and EDC strategies (Table 2).

Table 2 Summary statistics of aggregated portfolio returns.

portfolio strategy	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Classical Markowitz	-0.1082	-0.0263	-0.0015	-0.0024	0.0155	0.1126
Combined Markowitz	-0.0953	-0.0129	0.0017	0.0016	0.0148	0.1026
Classical EDC	-0.1770	-0.0295	0.0001	0.0009	0.0247	0.1389
Combined EDC	-0.1878	-0.0416	-0.0059	-0.0023	0.0418	0.1540

Figure 5 shows out-of-sample portfolio returns computed using the portfolio values out of the windows used to compute the portfolio weights, for different strategies and for each time window. The code is as follows.

```
sample_M <- NULL ## Markowitz
sample_C <- NULL ## Combined Markowitz
sample_EDC <- NULL ## Combined EDC
sample_mod_EDC<- NULL ## Combined EDC
icont <- 0
count <- 1:15
for (t in count){
  for(j in 1:125){
  icont = icont + 1;
  sample_M[icont] =
  ptf_results_M$ptf.oos.value[[t]][1,
  j]
  sample_C[icont] =
  ptf_results_C_M$ptf.oos.value[[t]][1,
  j]
  sample_EDC[icont] =
  ptf_results_EDC$ptf.oos.value[[t]][1,
  j]
  sample_mod_EDC[icont] =
  ptf_results_C_EDC$ptf.oos.value[[t]][1,
  j]
  }
}
```

```
date <- as.Date(sample.data.ts$Date,
        format = '%m/%d/%Y')
date_parz = seq(from = 1291, to = 3165,
by = 64)
m <- length(date_parz)
date_parz[m] = 3165
date_1 <- date[1291:3165]
date_2 <- date[date_parz]
dev.new()
par(mfrow = c(2, 1))
matplot(date_1, cbind(sample_M,
sample_C),
    type = 'l', col = c('red', 'black'),
    lty = c(2, 3), ylab =
    'profit & loss',
    xlab = '',xaxt='n',
    ylim = c(-0.20, 0.20),
    cex.lab = 1.2)
axis(1, date_2, format(date_2, '%m/%Y'),
    cex.axis = .9, las = 2)
legend('bottomright',
    legend = c('Classical Markowitz',
    'Combined Markowitz'),
    col=c('red','black'),lty = c(2, 3))
matplot(date_1, cbind(sample_EDC,
    sample_mod_EDC), type = 'l',
    col = c('brown', 'blue'),
    lty = c(3,1), ylab =
    'profit & loss',
    xlab = '', xaxt = 'n',
    ylim = c(-0.20, 0.20),
    cex.lab = 1.2)
axis(1, date_2,
     format(date_2, '%m/%Y'),
cex.axis = .9, las = 2)
legend('bottomright',
    legend =
    c('Classical EDC',
    'Combined EDC'),
    col = c('brown','blue'),
    lty = c(3, 1))
```

The classical Markowitz strategy faces several losses even when the combined one is able to make a profit (Figure 5 upper panel). This is especially evident during the negative peak in 2015, likely related to the oil crisis. This situation does not change when we compare classical and combined EDC portfolio strategies (Figure 5 lower panel).

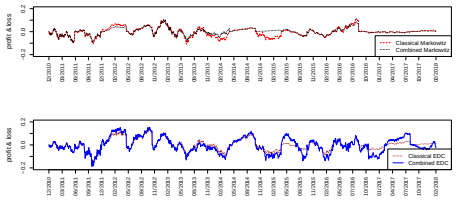

Fig. 5 Profit and loss curve for different strategies: Markowitz vs Combined Markowitz (upper panel) and EDC vs Combined EDC (lower panel).

5 Conclusions

This paper introduces the R package INFOSET for (i) modelling asset distribution detecting the parameters which describe left tail behaviour, (ii) clustering, (iii) labeling of the financial series for predictive and classification purposes and (iv) portfolio construction. We provide a review of the informative set and illustrate the uses and functionalities of the package. In particular, an empirical analysis is conducted on financial series that shows the effectiveness of this tool in carrying out its tasks.

6 Availability

The here presented package is written using S4 classes and provides methodology, summary and print to analyze the results. The R package INFOSET is available from the Comprehensive R Archive Network (CRAN) at https://CRAN.R-project.org/package=INFOSET.

References

Agarwalla SK, Varma JR, Virmani V (2021) The impact of covid-19 on tail risk: Evidence from nifty index options. Economics Letters 204:109,878

Ahelegbey DF, Giudici P, Mojtahedi F (2021) Tail risk measurement in crypto-asset markets. International Review of Financial Analysis 73:101,604

Artzner P, Delbaen F, Eber JM, et al (1999) Coherent measures of risk. Mathematical finance 9(3):203–228

Basalto N, Bellotti R, De Carlo F, et al (2007) Hausdorff clustering of financial time series. Physica A: Statistical Mechanics and its Applications 379(2):635–644

Chauveau D, Garel B, Mercier S (2019) Testing for univariate two-component gaussian mixture in practice. Journal de la Société Française de Statistique 160(1):86–113

Chen J, Li P (2009) Hypothesis test for normal mixture models: the em approach. The Annals of Statistics 37(5A):2523–2542

Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the em algorithm. Journal of the Royal Statistical Society Series B 39(1):1–38

Esling P, Agon C (2012) Time-series data mining. ACM Computing Surveys (CSUR) 45(1):1–34

Gao GP, Lu X, Song Z (2019) Tail risk concerns everywhere. Management Science 65(7):3111–3130

Giudici P, Polinesi G (2021) Crypto price discovery through correlation networks. Annals of Operations Research 299(1):443–457

Giudici P, Polinesi G, Spelta A (2022) Network models to improve robot advisory portfolios. Annals of Operations Research 313(2):965–989

Gkillas K, Longin F (2018) Financial market activity under capital controls: Lessons from extreme events. Economics Letters 171:10–13

Harris RD, Nguyen LH, Stoja E (2019) Systematic extreme downside risk. Journal of International Financial Markets, Institutions and Money 61:128–142

Kapadia N, Du J (2011) The tail in the volatility index. In: Fifth Singapore International Conference on Finance

Liao TW (2005) Clustering of time series data—a survey. Pattern recognition 38(11):1857–1874

MacDonald A, Scarrott CJ, Lee D, et al (2011) A flexible extreme value mixture model. Computational Statistics & Data Analysis 55(6):2137–2157

Mantegna RN (1999) Hierarchical structure in financial markets. The European Physical Journal B-Condensed Matter and Complex Systems 11:193–197

Mariani F, Ciommi M, Chelli FM, et al (2022a) An iterative approach to stratification: Poverty at regional level in italy. Social Indicators Research 161(2-3):873–903

Mariani F, Polinesi G, Recchioni MC (2022b) A tail-revisited markowitz mean-variance approach and a portfolio network centrality. Computational Management Science pp 1–31

Orlowski LT (2012) Financial crisis and extreme market risks: Evidence from europe. Review of Financial Economics 21(3):120–130

Rossi E, De Magistris PS (2013) Long memory and tail dependence in trading volume and volatility. Journal of Empirical Finance 22:94–112

Vlassis N, Likas A (1999) A kurtosis-based dynamic approach to gaussian mixture modeling. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans 29(4):393–399

West M (1993) Mixture models, monte carlo, bayesian updating, and dynamic models. Computing Science and Statistics pp 325–325

Yang ZR, Chen S (1998) Robust maximum likelihood training of heteroscedastic probabilistic neural networks. Neural Networks 11(4):739–747