Clustering market regimes using the Wasserstein distance [1]

Seminar for the Phd course in Quantitative Finance (SNS)

Alessandro Batignani

January 26, 2025

Link for the code: https://github.com/alebati3

Clusters analysis

- Cluster analysis or Clustering is an unsupervised technique used to group *objects* into *clusters*.
- "The definition of an optimal clustering is not well defined, and in the case of financial data, this is certainly true." [1]
- It's fundamental to define a way to quantify the similarity among objects.
- "Heuristically, we would like individual clusters to contain objects that are similar to each other whilst being distinct from objects in other clusters". [1]

k-means algorithm

- Suppose $X = \{(x_1, \ldots, x_N) : x_i \in V\}$, where $(V, \|\cdot\|_V)$ is a normed vector space . Each $x_i = (x_i^1, \ldots, x_i^d)$ is assumed to be standardized coordinate-wise, that is,

$$\mathbb{E}[(x_i^j)_{1\leq i\leq N}] = 0 \quad \text{and} \quad \mathrm{Var}((x_i^j)_{1\leq i\leq N}) = 1 \quad \text{for } j = 1, \ldots, d.$$

- The *k-means clustering algorithm* assigns elements of X to k disjoint clusters. Each of these clusters is defined by central elements $\bar{x} = \{\bar{x}_j\}_{j=1,\ldots,k}$ called *centroids*.

k-means algorithm

- Initially centroids are randomly sampled from X.
- At each step $n \in \mathbb{N}$ of the algorithm, one first calculates the *nearest neighbours*

$$C_l^n := \left\{ x_i \in X : \arg \min_{j=1,\dots,k} d(x_i, \bar{x}_j^{n-1}) = l \right\}$$

  associated to each $\bar{x}_l^{n-1}$ for $l = 1, \dots, k$.
- Each set C_l^n is then aggregated into a new centroid $\bar{x}_l^n$ for $l = 1, \dots, k$ via a function $\alpha : 2^V \to V$, so

$$\bar{x}_l^n := \alpha(C_l^n) \quad \text{for } l = 1, \dots, k.$$

- In the classical k-means on $\mathbb{R}^d$, we take as new centroid the barycenter of C_l

$$\alpha(C_l) = \left(\frac{1}{|C_l|} \sum_{x_j \in C_l} x_j \right)_{1 \le j \le d}.$$

  where, $|C_l|$ denotes the cardinality of the set C_l.

k-means algorithm

- For a given tolerance level $\epsilon > 0$ and a loss function $l : V^k \times V^k \to [0, +\infty)$, the k-means algorithm terminates at step $n^* \in \mathbb{N}$ if the stopping condition

$$l(\bar{x}^{n^*}, \bar{x}^{n^*-1}) < \epsilon$$

  is satisfied.
- The loss function l is given by

$$l(x, y) = \sum_{i=1}^{k} \|x_i - y_i\|_V,$$

- At the end, the algorithm outputs the final clusters $C^* = \{C_l^n\}_{l=1,\dots,k}$ and their k centroids $\bar{x}^n = \{x_l^n\}_{l=1,\dots,k}$.

The market regime clustering problem (MRCP)

Given the return series of a security price $\mathbf{r} = (r_0, r_1, \dots, r_N)$

- The MRCP is defined as the task of clustering segments of return series $(l_i)_{i=1}^{M}$, where

$$l_i = \left(r_i^1, \dots, r_i^n\right) \quad \text{for} \quad n \in \mathbb{N}$$

- Any vector $l_i \in \mathbb{R}^n$ can be associated to an empirical probability measure

$$\mu_i = \frac{1}{n} \sum_{j=1}^{n} \delta_{r_i^j}$$

  for $i = 1, \dots, M$ with n atoms.

- Thus the problem of clustering market regimes is equivalent to assigning a label to empirical probability measures $(\mu_i)_{i=1}^{M}$.

Problem setting and notation

Given the return series $\mathbf{r} = (r_j)_{j=0}^{N-1}$, where $r_j = \log(s_{j+1}) - \log(s_j)$, the segments of the return series are defined as follows:

- if $h_1, h_2 \in \mathbb{N}$ with $h_1 > h_2$ then

$$l_i = \left(r_{(h_1-h_2)(i-1)}, \ldots, r_{(h_1-h_2)(i-1)+h_1}\right) \quad \text{for} \quad i = 1, \ldots, M$$

  where M is the maximum number of partitions that can be extracted with the previous rule from the return series;
- every l_i has length h_1+1;
- h_2 is the sliding offset parameter:
  - It permits overlaps among partitions;
  - $h_2=0$ means no overlaps.

p-Wasserstein distance (W_p)

Main properties:

- W_p is a metric in the set of probability measures having the first p moments finite, denoted by $\mathcal{P}_p(\mathbb{R}^d)$.
- Convergence with respect to W_p is equivalent to the usual weak convergence of measures plus convergence of the first p moments.

W_p for empirical probability measure

- Suppose $\mu, \nu \in \mathcal{P}_p(\mathbb{R}^d)$ and let $d = 1$. Moreover, suppose that μ, ν are absolutely continuous with respect to the Lebesgue measure on $\mathbb{R}$. Then, the p-Wasserstein distance $W_p(\mu, \nu)$ is given by

$$W_p(\mu, \nu) = \left(\int_0^1 \left| F_\mu^{-1}(z) - F_\nu^{-1}(z) \right|^p dz \right)^{1/p},$$

  where the quantile function $F_\mu^{-1} : [0, 1) \to \mathbb{R}$ is defined as

$$F_\mu^{-1}(z) = \inf\{x : F_\mu(x) \geq z\}.$$

- If μ, ν are empirical measures with equal numbers of atoms $N \in \mathbb{N}$, with $(\alpha_i)_{1 \leq i \leq N}$ and $(\beta_i)_{1 \leq i \leq N}$ their corresponding order statistics, then

$$W_p(\mu, \nu)^p = \frac{1}{N} \sum_{i=1}^{N} |\alpha_i - \beta_i|^p .$$

W_p for empirical probability measure

- Suppose that $\{\mu_i\}_{1\leq i\leq M}$ are a family of empirical probability measures, each with order statistics $\{\alpha_j^i\}_{1\leq j\leq N}$.
- The Wasserstein barycenter is defined as the probability measure $\bar{\mu}$ that minimizes the sum of p-Wasserstein distances to each μ_i.
- In particular $\bar{\mu}$ is charaterazed by the following the order statistics

$$a_j = \text{Median}\left(\alpha_j^1, \ldots, \alpha_j^M\right) \quad \text{for } j = 1, \ldots, N.$$

Wasserstein k-means algorithm

- **Set of objects**: $\mathcal{K} = \{\mu_1, \ldots, \mu_M\}$;
- **Distance**: p-Wasserstein distance.
- **Aggregation function to update centroids**: Wasserstein barycenter.

The last specification to make is regarding the **loss function**:

- the most natural choice is to replace the distance induced by the norm on V with p-Wasserstein distance

$$l(\bar{\mu}^{n-1}, \bar{\mu}^n) = \sum_{i=1}^{k} W_p(\bar{\mu}_i^{n-1}, \bar{\mu}_i^n).$$

where $\bar{\mu}^n = (\bar{\mu}_i^n)_{1 \leq i \leq k}$ are the centroids obtained after step n of the Wasserstein k-means algorithm.

Alternative clustering algorithms as benchmarks

- k-means with statistical moments (Moment k-means)
- Hidden Markov model
  - HMM does not cluster segments of return series; instead, it associates to each log return a given latent state.
  - Emission probability densities are assumed to be gaussians (Gaussian HMM).

Moment k-means

- A natural and more classical approach to clustering regimes may involve studying the first $p \in \mathbb{N}$ raw moments associated to each measure $\mu \in \mathcal{K}$
- each empirical probability measure μ_i is mapped in vector of $\mathbb{R}^p$, whose components are the corresponding first p moments

$$\varphi^p(\mu_i) = \left(\int_{\mathbb{R}} x^n \mu_i(dx) \right)_{1 \leq n \leq p},$$

- Thus, for a given $p \geq 1$, we obtain

$$\varphi^p(\mathcal{K}) = \{\varphi^p(\mu_1), \ldots, \varphi^p(\mu_M) : \varphi^p(\mu_i) \in \mathbb{R}^p \text{ for } i = 1, \ldots, M\}.$$

- After standardising each element of $\varphi^p(\mathcal{K})$ component-wise, one can apply the standard k-means algorithm to this new set on $\mathbb{R}^p$.

Clustering validation on synthetic data

The generation of a synthetic price path facilitates the definition of a validation procedure.

- In this setting, every detail about price path and regime change periods is known;
  - It is possible to define scores to evaluate the performance of the algorithm.

Two different regimes are assumed:

- a standard regime (regime-off);
- the regime change (regime-on);

Clustering validation on synthetic data

Consider a time interval $[0, T]$, where $T \in \mathbb{N}$ represents the number of trading years.

- A mesh is created such that each time increment Δt roughly represents 1 trading hour:
  - $\Delta t = \frac{1}{n}$, with $n = 252 \times 7$;
- Next, the number of regime changes $r \in \mathbb{N}$ to be observed is defined;
  - one needs to specify the starting points and the length of each interval.

- **Simulation Note**: Price paths are generated over $\mathrm{T} = 20$ years with $\mathrm{r} = 10$ regime changes, randomly chosen with a duration of 0.5 years.

Example of synthetic path price

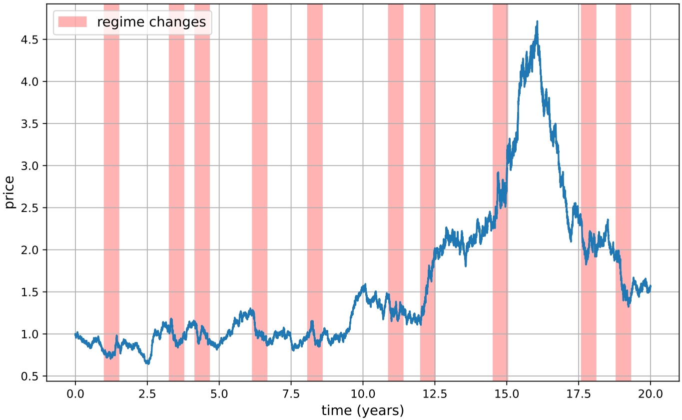


Figure: Synthetic geometric Brownian motion path with regime changes highlighted.

Clustering validation on synthetic data

- Each log-return r_i is a member of a set of $v_i \in \mathbb{N}$ empirical probability measures, $M_i = \{\mu_{j(i)}, \dots, \mu_{j(i)+v_i-1}\}$
  - where $j(i) \in \mathbb{N}$ is the first measure that r_i is a member of.
- Each measure in M_i is mapped to its corresponding predicted cluster labels, $\bar{y}^i = \{\bar{k}_{j(i)}, \dots, \bar{k}_{j(i)+v_i-1}\}$.
- Finally, these labels are aggregated into the vector

$$\bar{Y}^i = (\bar{Y}^i_0, \bar{Y}^i_1) = (\#\text{off-regime labels}, \#\text{on-regime labels})$$

for $i = 0, \dots, N-1$.

Accuracy scores

For a given vector of log-returns $\mathbf{r}$ and cluster assignments $C = \{C_l\}_{l=0}^{1}$, the followig scores are defined:

- **regime-off accuracy score (ROFS)**

$$\mathrm{ROFS}(\mathbf{r}, C) = \frac{\sum_{r_i \in off} \bar{Y}_0^i}{\sum_{r_i \in off} \sum_{k=0,1} \bar{Y}_k^i} \in [0, 1]$$

- **regime-on accuracy score (RONS)**

$$\mathrm{RONS}(\mathbf{r}, C) = \frac{\sum_{r_s^i \in on} \bar{Y}_1^i}{\sum_{r_i \in on} \sum_{k=0,1} \bar{Y}_k^i} \in [0, 1]$$

- **total accuracy (TA)**

$$\mathrm{TA}(\mathbf{r}, C) = \frac{\sum_{r_i \in off} \bar{Y}_0^i + \sum_{r_i \in on} \bar{Y}_1^i}{\sum_{i=1}^{N-1} \sum_{k=0,1} \bar{Y}_k^i} \in [0, 1]$$

Models for generating price paths

- Geometric Brownian motion (GBM)
- Merton Jump Diffusion model (MJD)

Geometric Brownian motion (GBM)

- $gBm(\mu, \sigma)$ is specified by the following SDE:

$$dS_t = \mu S_t \, dt + \sigma S_t \, dW_t$$

- **Solution of SDE:**

$$S_t = S_0 \exp\left(\left(\mu - \frac{\sigma^2}{2}\right) t + \sigma W_t\right)$$

- **off-regime parameters**: $(\mu_0, \sigma_0) = (0.02, 0.2)$

- **on-regime parameters**: $(\mu_1, \sigma_1) = (-0.02, 0.3)$

Geometric Brownian motion (GBM)

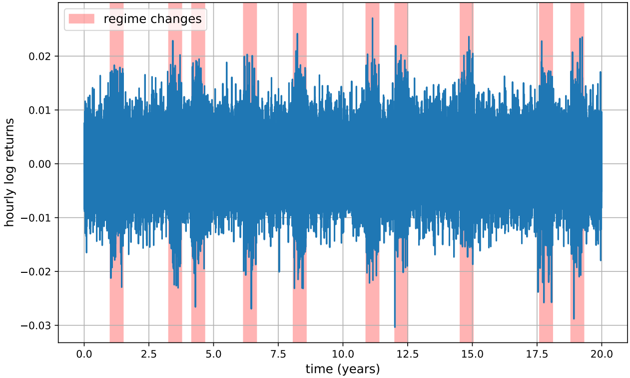


Figure: Plot of log returns associated with a synthetic geometric Brownian motion path, regime changes are highlighted.

W k-means on GBM data

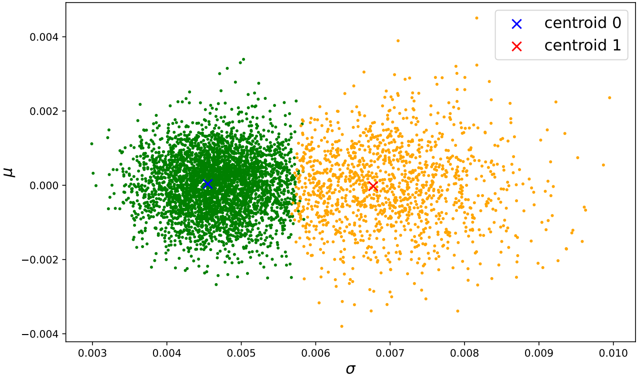


Figure: Plot of W K-means (h_1=35, h_2=28, p=1, tol=1e-08 and max_iter=600) clusters in mean-std space.

W k-means on GBM data

	RONS (%)	ROFS (%)	TA (%)	RUN TIME (s)
p = 1	mean = **92.89** CI = **(90.08, 95.05)**	mean = 96.34 CI = (94.57, 97.77)	mean = 95.47 CI = (94.25, 96.61)	mean = 2.86 CI = (2.71, 3.18)
p = 2	mean = **92.89** CI = **(90.08, 95.05)**	mean = 96.34 CI = (94.57, 97.81)	mean = 95.48 CI = (94.25, 96.61)	mean = 2.85 CI = (2.75, 2.99)
p = 3	mean = **92.89** CI = **(90.08, 95.05)**	mean = 96.34 CI = (94.57, 97.77)	mean = 95.48 CI = (94.25, 96.61)	mean = 2.80 CI = (2.71, 2.92)
p = 4	mean = 92.88 CI = (90.07, 95.05)	mean = 96.34 CI = (94.62, 97.77)	mean = 95.48 CI = (94.28, 96.61)	mean = 2.63 CI = (2.27, 2.95)
p = 20	mean = 92.88 CI = **(90.08, 95.05)**	mean = 96.34 CI = (94.57, 97.81)	mean = 95.47 CI = (94.25, 96.61)	mean = 2.35 CI = (2.22, 2.48)
p = 60	mean = 92.88 CI = (90.07, 95.05)	mean = 96.34 CI = (94.57, 97.81)	mean = 95.47 CI = (94.25, 96.61)	mean = 2.33 CI = (2.21, 2.46)
p = 100	mean = 74.19 CI = (7.05, 96.81)	mean = 82.20 CI = (43.76, 98.16)	mean = 80.20 CI = (36.07, 96.33)	mean = 2.29 CI = (2.18, 2.35)

Figure: Tabular with accuracy scores of W k-means (h_1=35, h_2=28, tol=1e-08 and max_iter=600) for different values of p.
95% CI are empirically calculated over 100 trials.

M k-means on GBM data

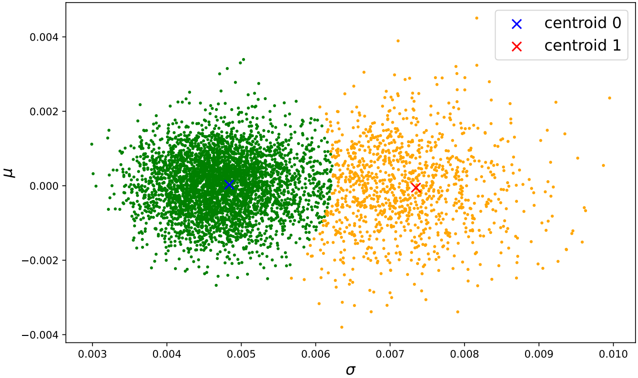


Figure: Plot of M K-means (h_1=35, h_2=28, p=2, tol=1e-08 and max_iter=600) clusters in mean-std space.

M k-means on GBM data

	RONS (%)	ROFS (%)	TA (%)	RUN TIME (s)
p = 2	mean = **77.60** CI = (48.70, 88.08)	mean = 89.13 CI = (50.96, 99.45)	mean = 86.24 CI = (50.58, 96.36)	mean = 2.85 CI = (2.68, 3.08)
p = 3	mean = 52.48 CI = (45.51, 71.78)	mean = 61.03 CI = (49.87, 95.57)	mean = 58.89 CI = (49.53, 88.86)	mean = 2.80 CI = (2.67, 2.96)
p = 4	mean = 75.93 CI = **(70.36, 81.19)**	mean = 99.43 CI = (99.00, 99.71)	mean = 93.55 CI = (92.15, 94.81)	mean = 2.88 CI = (2.72, 3.16)
p = 5	mean = 61.49 CI = (35.62, 80.51)	mean = 98.84 CI = (98.22, 99.70)	mean = 89.50 CI = (82.98, 94.60)	mean = 2.83 CI = (2.71, 3.03)
p = 6	mean = 63.77 CI = (49.33, 72.52)	mean = 99.60 CI = (99.20, 99.82)	mean = 90.63 CI = (87.05, 92.80)	mean = 2.82 CI = (2.72, 2.98)
p = 20	mean = 7.83 CI = (0.40, 55.24)	mean = 99.61 CI = (99.74, 100)	mean = 76.64 CI = (75.07, 88.57)	mean = 2.39 CI = (2.29, 2.47)
p = 100	mean = 2.21 CI = (0.39, 14.14)	mean = 99.99 CI = (99.92 ,100.0)	mean = 75.52 CI = (75.07, 78.46)	mean = 2.98 CI = (2.90, 3.06)

Figure: Tabular with accuracy scores of M k-means (h_1=35, h_2=28, tol=1e-08 and max_iter=600) for different values of p.
95% CI are empirically calculated over 100 trials.

Hidden Markov Model on GBM data

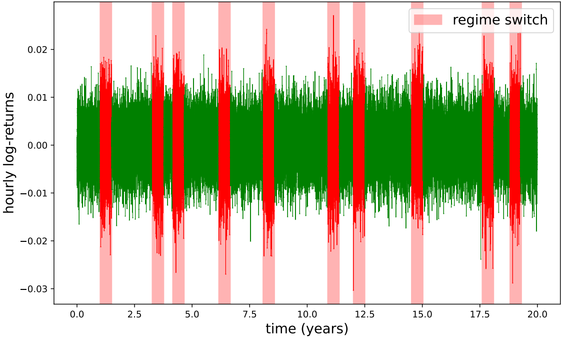


Figure: Plot of log returns classified with HMM (tol=1e-08 and max_iter=800).

GBM results - Summary

	RONS (%)	ROFS (%)	TA (%)	RUN TIME (s)
W k-means p = 1 tol = $1\text{x}10^{-8}$ max_iter = 600	mean = 92.89 CI = **(90.08, 95.05)**	mean = 96.34 CI = (94.57, 97.77)	mean = 95.47 CI = (94.25, 96.61)	mean = 2.86 CI = (2.71, 3.18)
M k-means p = 2 tol = $1\text{x}10^{-8}$ max_iter = 600	mean = 75.93 CI = (70.36, 81.19)	mean = 99.43 CI = (99.00, 99.71)	mean = 93.55 CI = (92.15, 94.81)	mean = 2.88 CI = (2.72, 3.16)
HMM tol = $1\text{x}10^{-8}$ max_iter = 800	mean = **93.11** CI = (1.04, 99.28)	mean = 99.27 CI = (99.21, 99.98)	mean = 97.74 CI = (75.00, 99.70)	mean = 3.27 CI = (0.33, 14.77)

Algorithm	Total	Regime-on	Regime-off	Runtime
Wasserstein	$90.60\% \pm 5.81\%$	$\mathbf{87.24}\% \pm 4.11\%$	$91.72\% \pm 6.46\%$	$0.87s \pm 0.16s$
Moment	$\mathbf{93.23}\% \pm 0.41\%$	$74.83\% \pm 1.57\%$	$\mathbf{99.38}\% \pm 0.1\%$	$1.06s \pm 0.16s$
HMM	$58.16\% \pm 7.11\%$	$41.51\% \pm 7.43\%$	$63.72\% \pm 11.94\%$	$0.58s \pm 0.36s$

Figure: [Top] Accuracy scores with 95% confidence intervals on synthetic gBm paths. CI are empirically calculated over 100 trials. For W and M k-means $h_1{=}35$ and $h_2{=}28$. [Bottom] Accuracy scores on sythetic gBm paths from [1].

Merton Jump-Diffusion Model (MJD)

- $MJD(\mu, \sigma, \lambda, \gamma, \delta)$ can be specified by the following SDE:

$$dS_t = \mu S_t \, dt + \sigma S_t \, dW_t + (J-1) S_t \, dN_t$$

- The arrival of the jumps is modelled by:

$$dN_t = \begin{cases} 1 & \text{with probability } \lambda \, dt \\ 0 & \text{with probability } 1 - \lambda \, dt \end{cases}$$

- Jump size is modelled by:

$$\log(J) = Y \sim \mathcal{N}(\gamma, \sigma^2)$$

- **Solution of SDE:**

$$S_t = S_0 \exp\left(\left(\mu - \frac{\sigma^2}{2} \right) t + \sigma W_t + \sum_{j=1}^{N(t)} Y_j \right)$$

Merton Jump-Diffusion Model (MJD)

- **off-regime parameters:**
  $(\mu_0, \sigma_0, \lambda_0, \gamma_0, \delta_0) = (0.05,\ 0.2,\ 5,\ 0.02,\ 0.0125)$
- **on-regime parameters:**
  $(\mu_1, \sigma_1, \lambda_1, \gamma_1, \delta_1) = (-0.05,\ 0.4,\ 10,\ -0.04,\ 0.1)$

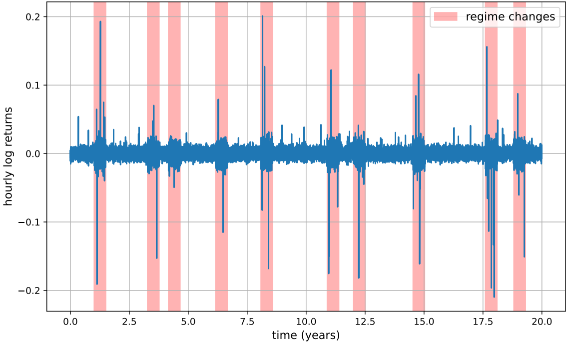


Figure: Plot of log returns associated with a synthetic Merton jump diffusion path, regime changes are highlighted.

W k-means on MJD data

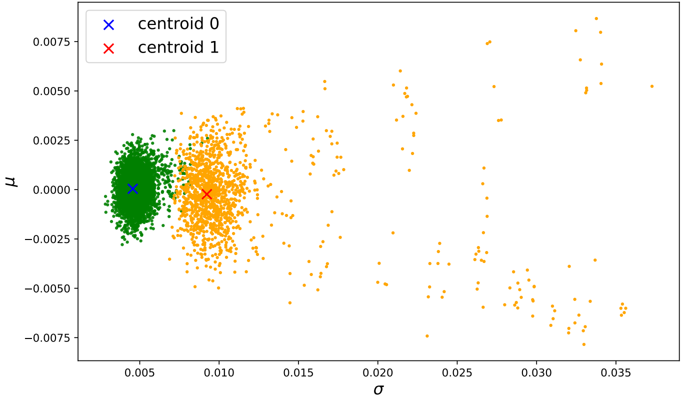


Figure: Plot of W K-means (h_1=35, h_2=28, p=1, tol=1e-08 and max_iter=600) clusters in the mean-std space.

W k-means on MJD data

	RONS (%)	ROFS (%)	TA (%)	RUN TIME (s)
p = 1	mean = **96.26** CI = **(96.84, 98.77)**	mean = 98.72 CI = (97.94, 99.32)	mean = 98.10 CI = (97.86, 98.99)	mean = 2.42 CI = (2.19, 2.71)
p = 2	mean = 93.50 CI = (9.21, 98.73)	mean = 98.75 CI = (97.94, 99.92)	mean = 97.44 CI = (77.23, 98.99)	mean = 2.31 CI = (2.19, 2.40)
p = 3	mean = 93.54 CI = (9.08, 98.73)	mean = 98.75 CI = (97.94, 99.96)	mean = 97.45 CI = (77.23, 98.99)	mean = 2.31 CI = (2.19, 2.37)
p = 4	mean = 94.48 CI = (12.77, 98.73)	mean = 98.75 CI = (97.94, 99.96)	mean = 97.68 CI = (78.17, 98.98)	mean = 2.31 CI = (2.19, 2.40)
p = 20	mean = 92.46 CI = (6.16, 98.77)	mean = 98.76 CI = (97.94, 99.99)	mean = 97.19 CI = (76.49, 98.99)	mean = 2.30 CI = (2.18, 2.37)
p = 60	mean = 93.44 CI = (7.56, 98.77)	mean = 98.77 CI = (97.94, 99.94)	mean = 97.43 CI = (76.80, 98.99)	mean = 2.29 CI = (2.19, 2.36)
p = 100	mean = 91.01 CI = (8.27, 98.85)	mean = 98.00 CI = (97.47, 99.99)	mean = 96.25 CI = (76.32, 98.98)	mean = 2.29 CI = (2.18, 2.37)

Figure: Tabular with accuracy scores of W k-means (h_1=35, h_2=28, tol=1e-08 and max_iter=600) for different values of p.
95% CI are empirically calculated over 100 trials.

MJD results - Summary

	RONS (%)	ROFS (%)	TA (%)	RUN TIME (s)
W k-means p = 1 tol = 1×10^{-8} max_iter = 600	mean = **96.26** CI = **(96.83, 98.77)**	mean = 98.72 CI = (97.94, 99.32)	mean = 98.10 CI = (97.86, 98.99)	mean = 2.42 CI = (2.19, 2.70)
M k-means p = 2 tol = 1×10^{-8} max_iter = 600	mean = 22.06 CI = (3.67, 52.29)	mean = 88.54 CI = (58.03, 100.0)	mean = 71.91 CI = (56.15, 76.97)	mean = 2.61 CI = (2.56, 2.69)
HMM tol = 1×10^{-8} max_iter = 800	mean = 95.28 CI = (86.10, 99.46)	mean = 99.71 CI = (99.49, 99.87)	mean = 98.60 CI = (96.23, 99.69)	mean = 2.06 CI = (0.83, 3.36)

Algorithm	Total	Regime-on	Regime-off	Runtime
Wasserstein	$\mathbf{91.28}\% \pm 4.08\%$	$\mathbf{86.87}\% \pm 3.1\%$	$92.76\% \pm 4.43\%$	$1.11s \pm 0.25s$
Moment	$66.64\% \pm 3.42\%$	$27.25\% \pm 8.73\%$	$79.79\% \pm 7.40\%$	$1.71s \pm 0.28s$
HMM	$75.05\% \pm 0.01\%$	$0.66\% \pm 0.04\%$	$\mathbf{99.87}\% \pm 0.01\%$	$0.66s \pm 0.04s$

Figure: [Top] Accuracy scores with 95% confidence intervals on MJD synthetic paths. CI are empirically calculated over 100 trials. For W and M k-means h_1=35 and h_2=28. [Bottom] Accuracy scores on synthetic Merton jump diffusion paths from [1].

Clustering validation on real data

- Clusters derived using k-means are typically evaluated using the (average) silhouette score:
  - is a distance-based score in the range [-1, 1], that captures both internal cohesion of clusters and their degree of separation.
  - for values close to 1: each object is closer to objects within the same cluster than to those of other clusters;
- Since the silhouette score depends on the distance between objects, is not fair to compare clusterings referred to different distances.

Maximum mean discrepancy (MMD)

- Let $(\mathcal{X}, d)$ be a metric space and $\mathcal{F}$ be a class of functions $f : \mathcal{X} \to \mathbb{R}$. If $\mu, \nu \in \mathcal{P}(\mathcal{X})$ are Borel measures, the **maximum mean discrepancy (MMD)** between μ and ν is defined as

$$\mathrm{MMD}[\mathcal{F}, \mu, \nu] := \sup_{f \in \mathcal{F}} \left(\mathbb{E}_\mu[f(x)] - \mathbb{E}_\nu[f(y)] \right).$$

- If $\mathcal{F}$ is the Gaussian kernel

$$\kappa_G : \mathbb{R}^d \times \mathbb{R}^d \to [0, +\infty), \quad \kappa_G(x, y) = \exp\left(-\frac{\|x - y\|^2_{\mathbb{R}^d}}{2\sigma^2} \right)$$

  then the MMD is a metric on $\mathcal{P}(\mathcal{X})$.

- **Note**: in the subsequent simulations, a gaussian kernel κ_G is chosen with σ=0.1.

Maximum Mean Discrepancy (MMD)

- If μ and ν are empirical probability measures, associated with the populations $(x_1, \ldots, x_n)$ and $(y_1, \ldots, y_m)$, the MMD is computed by:

$$\mathrm{MMD}^2[\kappa_G, \mu, \nu] = \left[\frac{1}{n^2} \sum_{i,j=1}^{n} k_G(x_i, x_j) - \frac{2}{mn} \sum_{i,j=1}^{m,n} k_G(x_i, y_j) \right.$$
$$\left. + \frac{1}{m^2} \sum_{i,j=1}^{m} k_G(y_i, y_j) \right].$$

Cluster validation via MMD

Between-cluster evaluation

- given the two cluster C_0, C_1, draw $n \in \mathbb{N}$ empirical probability measure pairs $(\mu_i, \nu_i) \in C_0 \times C_1$ for $i = 1, \dots, n$.
- For each pair, compute $\mathrm{MMD}^2[\kappa_G, \mu_i, \nu_i]$.
- Finally, the between-cluster similarity score is defined as

$$\mathrm{bSim} = \mathrm{Median}\left(\left(\mathrm{MMD}^2[\kappa_G, \mu_i, \nu_i] \right)_{1 \leq i \leq n} \right),$$

Cluster validation via MMD

Within-cluster evaluation

- for each cluster C_l, $l = 0, 1$, we draw $n \in \mathbb{N}$ empirical probability measure pairs $(\mu_i^0, \mu_i^1) \in C_l \times C_l$
  - for each pair, compute $\mathrm{MMD}^2[\kappa_G, \mu_i^0, \mu_i^1]$.
  - the within-cluster similarity score is defined as

$$\mathrm{wSim}_l = \mathrm{Median}\left(\left(\mathrm{MMD}^2[\kappa_G, \mu_i, \nu_i]\right)_{1 \leq i \leq n}\right),$$

- **Simulation Notes**: the number of pairs, n, is set to 100,000 for the subsequent simulations.

IBM data

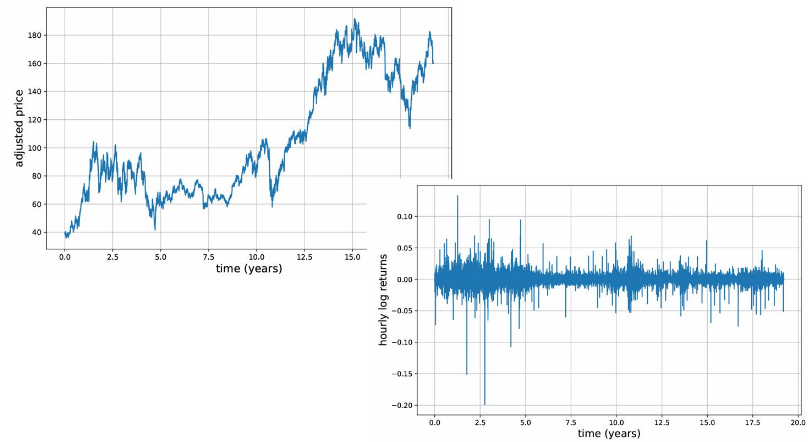


Figure: Hourly IBM Data from January 2, 1998, to April 28, 2017. Adjusted path price for IBM [top left], and associated log returns [bottom right].

W k-means on IBM data.

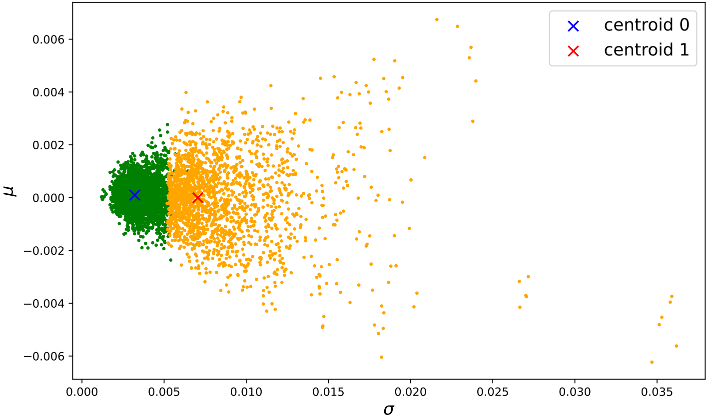


Figure: Plot of W k-means (h_1=35, h_2=28, p=1, tol=1e-08 and max_iter=600) clusters in the mean-std space.

M k-means on IBM data.

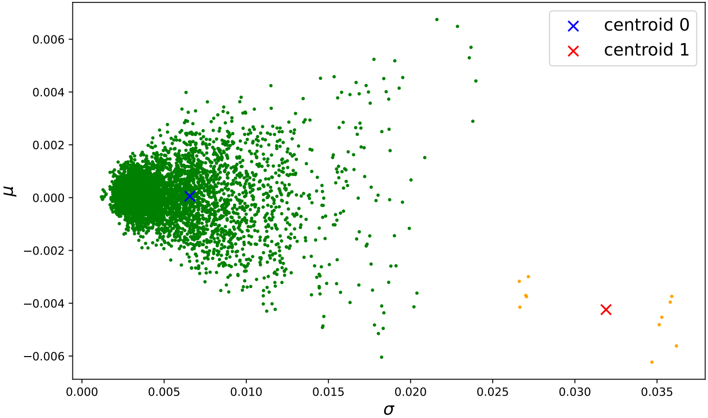


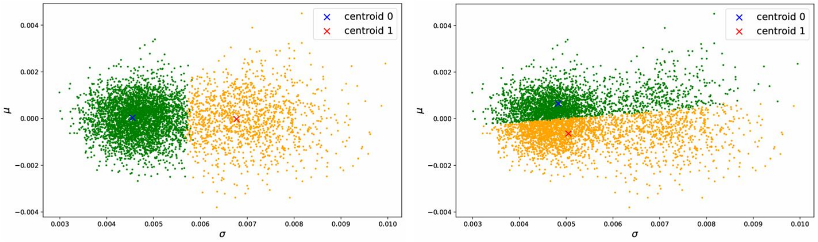
Figure: Plot of M k-means (h_1=35, h_2=28, p=4, tol=1e-08 and max_iter=600) clusters in the mean-std space.

Clustering validation for IBM data

	bSIM	wSIM_off	wSIM_on
W k-means p = 1 tol = $1\text{x}10^{-8}$ max_iter = 600	mean = 1.49e-04 CI = (1.47e-04, 1.51e-04)	mean = **3.33e-05** CI = (3.28e-05, 3.36e-05)	mean = 2.27e-04 CI = (2.24e-04, 2.30e-04)
M k-means p = 4 tol = $1\text{x}10^{-8}$ max_iter = 600	mean = **1.81e-03** CI = (1.80e-03, 1.82e-03)	mean = 9.53e-05 CI = (9.41e-05, 9.64e-05)	mean = **1.88e-04**

Figure: Clustering validation scores with 95% confidence intervals for IBM data using MMD. CI are empirically calculated over 100 trials. For W and M k-means h_1=35 and h_2=28.

Validation via MMD vs accuracy scores on GBM data



	RONS (%)	ROFS (%)	TA (%)	bSIM	wSIM_off	wSIM_on
W k-means p = 1 tol = 1×10^{-8} max_iter = 600	**94.38**	**96.08**	**95.65**	9.07e-05	5.85e-05	1.11e-04
W k-means p = 1 tol = 1×10^{-4} max_iter = 300	62.82	54.31	56.44	**1.63e-04**	**2.47e-05**	**3.08e-05**

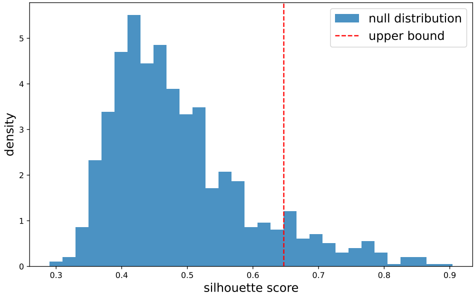
Figure: Plots of W k-means (h_1=35, h_2=28) in the mean-std space with tol=1e-08 [top left] and tol=1e-04 [top right]; associated tabular with accuracy scores and cluster similarity indexes [bottom].

Are the clusters found really significant?

- **Problem:** almost every clustering algorithm will find clusters in a data, even if that data has no natural cluster structure.
- Statistical testing procedures provide a useful method to assess the significance of clusters that have been discovered.
  - In particular, one can test the null hypothesis that no cluster structure exists among the instances.
- Right-tailed test
  - test statistics: a numerical value that summarize the clustering;
  - null distribution for the test statistics;
  - significance level;

Right-tailed test

- silhouette score as test statistics;
- to get a meaningful null distribution, one needs to generate data with overall properties and characteristics as similar as possible to real data except that it has no cluster structure;
- Given the null distribution and a significance level α, one can determine the upper bound of the non-critical region.



Null distribution

Choice of the Null Model:

- GARCH(1,1) with Gaussian conditional pdf.

Null distribution generation:

- fit a GARCH(1,1) with Gaussian conditional pdf to IBM data;
- generate 1000 series of log-returns from GARCH(1,1) with optimal parameters;
- for each series of log-returns execute a W k-means and compute the silhouette score.

Result of the right-tailed test

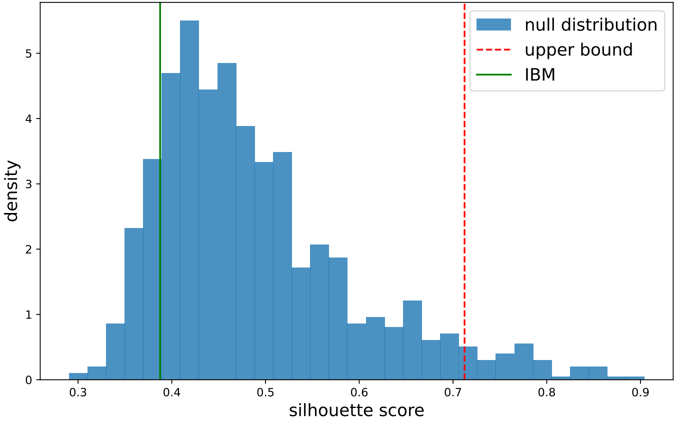


Figure: Result of the right-tailed test ($\alpha = 5\%$) for W k-means (h_1=35, h_2=28, p=1, tol=1e-08 and max_iter=600) on IBM data.

Conclusion

"The validation of clustering structures is the most difficult and frustrating part of cluster analysis.

Without a strong effort in this direction, cluster analysis will remain a black art accessible only to true believers who have experience and great courage. ' [3]

References

- Blanka Horvath, Zacharia Issa and Aitor Muguruza. Clustering market regimes using the Wasserstein distance. October 2021.
- James Mc Greevy et al. Detecting multivariate market regimes via clustering algorithms. March 2024.
- Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. Introduction to Data Mining. Pearson, 2018 (Second edition).
- Rosario N. Mantegna and H. Eugene Stanley. Introduction to Econophysics: Correlations and Complexity in Finance. Cambridge University Press, 2007.