République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Abderrahmane MIRA de BEJAIA
Faculté des Sciences Exactes
Département de Recherche Opérationnelle

Mémoire de Master

Présenté par

MIMOUNE Imane

Filière : Mathématiques Appliquées

Option : Mathématiques Financières

Thème

Clustering de Séries Chronologiques

Soutenue le : 09/07/2024 Devant le Jury composé de :

Nom et Prénom		
Karima BOUIBED	Université de Bejaia	**Présidente**
TOUATI Sofiane	Université de Bejaia	**Encadreur**
Zohra AOUDIA	Université de Bejaia	**Examinatrice**
Nacim NAIT-MOHAND	Université de Bejaia	**Examinateur**

Année universitaire : **2023/2024**

Remerciements

Avant tout, je souhaite exprimer ma gratitude envers Dieu tout-puissant pour m'avoir donné la force et la détermination nécessaires pour accomplir ce travail.

Je tiens à exprimer ma profonde gratitude à toutes les personnes qui ont contribué, de près ou de loin, à la réalisation de ce mémoire de fin de cycle.

Je suis profondément reconnaissante envers mon encadrant, Monsieur **Touati Sofiane** pour sa guidance éclairée, son soutien précieux et ses conseils avisés tout au long de cette recherche. Son expertise et sa patience ont été d'une aide inestimable.

Je tiens également à exprimer ma gratitude envers mes enseignants. Ils ont partagé leur savoir et leur passion, et ont toujours été disponibles pour répondre à mes questions.

Un grand merci à ma famille et à mes amis pour leur soutien moral indéfectible, leur compréhension tout au long de cette période exigeante, leur patience et leurs encouragements m'ont permis de mener à bien ce projet.

Merci à toutes et à tous.

Je dédie ce mémoire

À mes parents :

Mes chers parents, **MAOUCHE Louisa** et **MIMOUNE Boussaad**, pour leur soutien inconditionnel et leur amour indéfectible tout au long de ce parcours. Leur encouragement constant, leurs sacrifices et leur foi en mes capacités ont été des sources inestimables de motivation. Ce projet n'aurait jamais existé sans eux. Je leur suis reconnaissante pour tout ce qu'ils ont fait pour moi.

À mes frères :

À mes fréres **Mohand** et **Belkacem**, pour votre soutien indéfectible et vos encouragements constants.

À mes grands mères :

Mes chères grands-mères, je vous remercie pour votre amour inconditionnel, vos encouragements et votre sagesse.

À mes tante :

À mes tantes adorées, **Dalila** et **Samia**. Vous avez été des piliers de soutien, des fontaines de sagesse. Merci pour les moments de complicité, les encouragements chaleureux, pour votre présence réconfortante et votre influence positive

À mon grand-père et mon cousin :

À la mémoire de mon cher grand-père, **MAOUCHE Yahia**, et de mon cousin, **AIT MOHAMED Ayoub**. Bien que vous ne soyez plus parmi nous, vos souvenirs continuent de vivre en moi et de m'inspirer chaque jour. Merci pour votre amour, votre sagesse et les moments précieux que nous avons partagés. C'est avec une profonde gratitude que je vous rends hommage à travers ce travail.

À mes soeurs Amel, Amel, Maria, Racha, Imane, Zahra :

qui ont été à mes côtés dans les moments de doute et de difficulté. Votre soutien moral, vos encouragements et votre amitié m'ont permis de surmonter les obstacles et de persévérer. Je vous remercie sincèrement pour les précieux instants et les moments de joie et de partage.

À TOUATI Sofiane :

Je souhaite dédie ce mémoire à mon encadrant, en reconnaissance de son soutien et ses conseils et de sa patience tout au long de ce projet. Je vous remercie sincèrement pour votre engagement et votre dévouement.

À tous ceux qui ont contribué, de près ou de loin, à la réalisation de ce travail :

Je tiens également à exprimer ma gratitude envers toutes les personnes qui ont contribué, de près ou de loin, à la réalisation de ce mémoire. Vos conseils, votre soutien et votre précieuse aide ont permis la concrétisation de ce projet. Un grand merci à vous tous.

Table des matières

Liste des Algorithmes

Table des figures

Liste des tableaux

Introduction générale

Dans un monde de plus en plus axé sur les données, l'analyse et l'extraction d'informations significatives à partir de vastes ensembles de données sont devenues essentielles. L'objectif principale du traitement de données est de pouvoir en tirer une représentation simplifiée, permettant ainsi de comprendre le phénomène étudié. Néanmoins, la nature diverse et la grande quantité de données rend la tâche difficile. Parmi les différentes formes de données, les séries chronologiques occupent une place particulière en raison de leur capacité à capturer des dynamiques temporelles complexes. En fait, toutes données collectées l'est à un instant donné. Elles sont omniprésentes dans de nombreux domaines tels que la finance, la météorologie, la médecine, l'économie, et bien d'autres encore.

Le clustering de données est le problème consistant à regrouper des données dans des sous-ensembles, appelé Cluster, contenant des données *similaires*. Cette simple définition regroupe diverses manières de réaliser un tel Clustering, comme le Clustering partitionnel et le célèbre algorithme K-means permettant de le réaliser. Les méthodes de Clustering appliquées aux séries chronologiques ont permis d'apporter des solutions à des problèmes comme la détection de valeurs extrêmes, la segmentation des séries en laps de temps où la série est similaire, le regroupement dans des Clusters de séries chronologiques.

Ce présent mémoire s'inscrit dans la dernier type . Le Clustering de séries chronologiques est une technique qui permet de regrouper des séries similaires afin de révéler des structures sous-jacentes, de détecter des anomalies. Contrairement aux données statiques, les séries chronologiques présentent des défis spécifiques en raison de la dépendance temporelle et de la variation dynamique de leurs valeurs. Ce mémoire vise à explorer une partie de la littérature du clustering des séries chronologiques, en mettant l'accent sur les différentes mesures de similarité de séries chronologiques, ainsi que l'algorithmes de Clustering K-means. En deuxième lieu, d'évaluer les mesures de similarité en visualisant dans le plan la proximité de séries identiques ou proches, basé sur ces mesures. Enfin, nous présenterons une analyse comparative du K-means à travers des expériences empiriques sur des jeux de données réels.

Ce mémoire est partagé en trois chapitres et une conclusion générale :

Dans le **Chapitre 1 : Notions générales sur les séries chronologiques**, nous rappelons les notions de processus stochastiques appliqués aux séries chronologiques.

Dans le **Chapitre 2 : Clustering des séries chronologiques**, on traite des méthodes et des algorithmes de clustering appliqués aux séries chronologiques.

Le **Chapitre 3 : Application des méthodes de Clustering de séries temporelles simulées**, se concentre sur l'application pratique des méthodes de clustering sur des données simulées.

Enfin, nous terminons par une Conclusion Générale.

Chapitre 1

Notions générales sur les séries chronologiques

1.1 Introduction

Une série chronologique, également appelée série temporelle, désigne une suite de données arrangées dans l'ordre du temps. Ces données peuvent être recueillies de manière périodique, que ce soit quotidiennement, mensuellement, annuellement, ou à des intervalles irréguliers. Les séries chronologiques trouvent leur application dans divers domaines tels que l'économie, la finance, la météorologie, la santé publique..., permettant l'analyse des tendances, des modèles, et des variations au fil du temps.

Une série chronologique $Y = (Y_t)_{t\in T}$ est une collection de vecteurs aléatoires ayant des valeurs dans l'espace d'états $E = \mathbb{R}^k$, où $T \subseteq \mathbb{Z}$ est un ensemble d'entiers naturels ou relatifs. Si $E = \mathbb{R}$, le processus est qualifié de unidimensionnel (ou uni-varié). Un processus aléatoire uni-varié Y_t indexé par T est considéré de second ordre s'il satisfait les trois propriétés suivantes :

- **la stationnnarité au sens stricte** : la distribution conjointe des valeurs de la série ne change pas au fil du temps ;
- **la moyenne constante** : la moyenne $E(Y_t)$ est constante pour chaque instant t ;
- **une fonction d'auto-covariance absolument sommable** : la covariance $cov(Y_t, Y_s)$ existe et finie pour chaque paire (t, s) de T.

ce qui signifie que les moyennes $E(Y_t)$ et les variances et les covariances $cov(Y_t, Y_s) = E(Y_tY_s) - E(Y_t)E(Y_s)$ existent et sont finies pour chaque t, s de valeurs de T. Cette définition englobe le cas des séries qui ne sont pas régulièrement espacées dans le temps.

Dans la pratique, une observation d'un processus aléatoire sur une période finie est appelée série chronologique par abus de langage. Cette observation consiste à observer des valeurs discrètes d'une variable réelle ou complexe à des moments spécifiques. Prenat et. al. [2010]

Exemple 1.1. *Nous avons rassemblé des données sur les ventes mensuelles de voitures au cours des dernières années. Chaque observation représente le total des ventes de voitures pour un mois donné T. Cette série de donnée est un exemple de série univariée car elle se compose d'une seule variable, à savoir le nombre de voitures vendues, mesurée à différents moments dans le temps. L'espérance de cette série correspondrait à la valeur moyenne anticipée du nombre de ventes de voitures chaque mois, donc cette dernière existe et finie.*

Exemple 1.2. *Prenons l'exemple du nombre annuel de naissances. Chaque observation représente le total des naissances pour une année donnée. Nous pouvons inclure le genre (masculin ou féminin), l'âge de la mère comme variables et cela nous permettra d'analyser les différences entre les naissances de garçons et de filles et aussi comparer les naissances entre les mères plus jeunes et plus âgées. L'espérance de cette série multivariée serait la valeur*

attendue du nombre de naissances chaque année, en tenant compte des variables genre et âge de la mère (existe et finie). Nous pouvons étudier les variations d'une année à l'autre en fonction des variables multivariées.

L'analyse des séries temporelles est en effet un domaine essentiel pour comprendre et exploiter les données temporelles Messaouda [2020]. Voici quelques points concernant l'analyse des séries temporelles :

1. **Détection de tendances et de motifs :** l'objectif est de détecter les tendances à long terme, les cycles et les saisons dans les données temporelles. Cela nous permet de mieux comprendre le comportement de la série temporelle.
2. **Prévision et prédiction :** l'utilisation de modèles permet d'extrapoler et de prédire les valeurs futures d'une série temporelle.
3. **Analyse des corrélations et des relations :** explorer les relations entre différentes séries temporelles ou entre une série temporelle et d'autres variables.
4. **Diagnostics et détection d'anomalies :** détection des valeurs aberrantes ou des points atypiques dans la série temporelle.
5. **Prise de décision informée :** les informations extraites de l'analyse des séries temporelles aident à prendre des décisions éclairées.

On peut trouver des exemples de séries chronologiques univariées dans de nombreux domaines. Le tableau suivant donne des exemples :

Domaine	**Application**
Finance et économétrie	-évolution des indices boursiers, des ventes et achats de biens, des productions agricoles ou industrielles
Assurance	-analyse des sinistres
traitement du signal	-signaux de communications, de radars, de sonars, analyse de la parole
Médecine / biologie	-suivi des évolutions des pathologies -analyse d'électro-encéphalogrammes et d'électrocardiogrammes
métrologie	-variation de phase ou de fréquence des oscillateurs, dérive et bruit des capteurs inertiels.

Tableau 1.1 – Domaines d'applications des séries chronologiques. Tiré de SEDDATI [2019]

1.2 Notions fondamentales sur les séries chronologiques

Les séries chronologiques sont généralement constituées de divers éléments qui facilitent une compréhension approfondie de la structure fondamentale des données au fil du temps. De même des notions et outils de calcul permettent de préparer et de faire une première analyse des données chronologique Touche [2022]. Parmi ces éléments, on trouve :

1. **Tendance :** cette composante exprime la trajectoire à long terme de la série ou son évolution fondamentale au fil du temps.
2. **Saisonnalité :** la saisonnalité fait référence aux variations périodiques qui se produisent à des intervalles réguliers, pouvant être corrélées aux saisons, aux cycles économiques ou à d'autres motifs récurrents.

3. **Composante irrégulière (ou bruit) :** Cette composante regroupe les variations résiduelles ou accidentelles, manifestant des fluctuations irrégulières et imprévisibles. Elle constitue la partie aléatoire d'une série chronologique.

Exemple 1.3. *La Figure 1.1 montre les températures quotidienne sur une année. La série réelle en bleu est surélevée par la tendance par rapport à la série saisonale.*

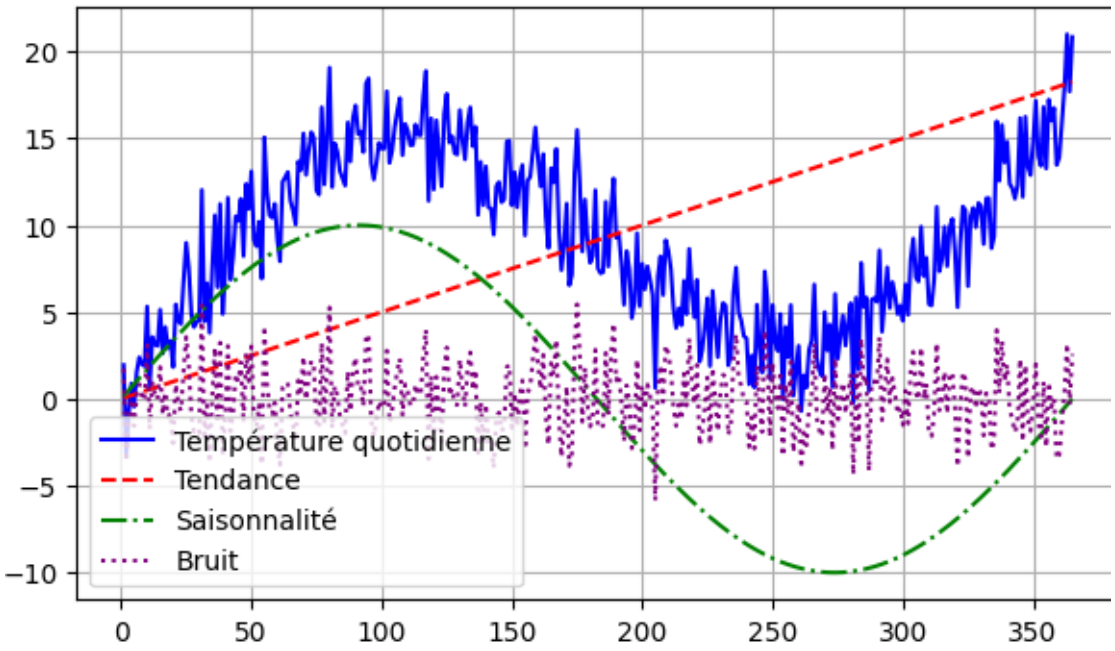

FIGURE 1.1 – Série temporelle de température quotidienne

Définition 1.1. *Un processus $(\varepsilon_t)_{t\in T}$ est dit bruit blanc s'il vérifie les conditions suivantes :*

1. $E(\varepsilon_t) = 0$
2. $var(\varepsilon_t) = \sigma^2$
3. $cov(\varepsilon_t, \varepsilon_{t-h}) = 0,\ \forall t \text{ et } h \neq 0$

Une propriété importante des séries chronologiques est la stationnarité. Un processus est dit stationnaire si ses propriétés statistiques (moyenne, variance, covariance) restent constantes dans le temps. On distingue deux types de stationnarité : la stationnarité faible et forte. Bigot [2017]

Définition 1.2. *Un processus $(Y_t)_{t\in T}$ est dit* ***faiblement stationnaire*** *si :*

1. $E(Y_t) = \mu \forall t \in T$, *où* μ *est constant ;*
2. $Var(Y_t) = \sigma^2 < \infty, \forall t \in T$, *où* σ *est constant ;*
3. $Cov(Y_t, Y_{t-h}) = \psi_h, \forall t, h \in T$.

Définition 1.3. *Un processus Y_t est considéré* ***fortement stationnaire*** *(ou strictement stationnaire) lorsque toute ses observations ont la même loi (la distribution conjointe de ses observations reste constante et invariante quelle que soit la période).*

On remarque que la stationnarité forte (stricte) implique la stationnarité faible. Ainsi, les caractéristiques d'un processus fortement stationnaire (moyenne, variance, covariance) observées à un instant donné sont les mêmes que celles observées à un instant h plus tard.

L'autocorrélation d'une série temporelle se réfère au fait que la mesure d'un phénomène à un instant t peut être corrélée avec les mesures précédentes $(t_1, t_2, t_3, ...)$. La fonction de

covariance (ou d'autocovariance) $\gamma(h)_{h\in T}$ mesure la covariance entre les variables Y_t et Y_{t-h}, pour un décalage h :

$$\gamma(h) = cov(Y_t, Y_{t-h}) = E[(Y_t - E(Y_t))(Y_{t-h} - E(Y_{t-h}))], t, h \in T. \tag{1.1}$$

Ainsi :

$$\gamma(0) = E[(Y_t - E(Y_t))^2] = V(Y_t) = \sigma_Y^2. \tag{1.2}$$

Dans le cas d'une série stationnaire, la fonction d'autocorrélation (ACF) $\rho(h)_{h\in T}$ est définie par :

$$\rho_h = \frac{\gamma_h}{\gamma_0}, h \in T; \tag{1.3}$$

La représentation graphique de la fonction autocorrélation ρ_h est appelée **Corrélogramme**.

L'autocorrélation partielle (PACF) mesure la corrélation entre les valeurs de la série temporelle séparées par k intervalles, en tenant compte des valeurs des intervalles intermédiaires. Elle est représentée par **corrélogramme partiel**.

Un autre outil d'analyse d'une série chronologique est celui du periodogramme. Le périodogramme est particulièrement utile pour identifier les cycles ou les périodicités dans les données Caiado et. al. [2006]. Soient $w_j = \frac{2\pi j}{n}$, $j = 1, \ldots, \left\lfloor \frac{n}{2} \right\rfloor$ des fréquences prises dans l'intervalle $[0, \pi]$. Le périodogramme d'une série $(x_t)_{t\in T}$ est donné par :

$$P_x(w_j) = \frac{1}{n}\left|\sum_{t=1}^{n} x_t e^{-itw_j}\right|^2, \ \forall j = 1, \ldots, \left\lfloor \frac{n}{2} \right\rfloor. \tag{1.4}$$

Définition 1.4. *On appelle* ***opérateur de retard*** *l'opérateur B qui transforme la série Y_t en Y_{t-1} tel que :*

$$BY_t = Y_{t-1}, t \in T \tag{1.5}$$

On peut répéter cette procédure de manière itérative et définir par récurrence

$$B^m Y_t = Y_{t-m}, m \in \mathbb{N} \tag{1.6}$$

Définition 1.5. *L'**opérateur de différenciation** Δ est utilisé pour calculer la différence entre les valeurs successives d'une série avec :*

$$\Delta Y_t = Y_t - Y_{t-1}, t \in T \tag{1.7}$$

Remarque 1.1. *On a $\Delta Y_t = Y_t - Y_{t-1} = Y_t - BY_t = (1-B)Y_t$, on en déduit que l'opérateur Δ est identique à $1 - B$.*

Dans le cas d'une présence de la saisonnalité on applique l'opérateur défini ci-dessous.

Définition 1.6. *Pour une série $(Y_t)_{t\in T}$, **l'opérateur de désaisonnalité** Δ_s est défini comme suit :*

$$\Delta_s Y_t = Y_t - Y_{t-s}, t \in T, \tag{1.8}$$

où s est la période de la saisonnalité.

Nous avons la relation suivante :

$$\Delta_s = (1 - B^s). \tag{1.9}$$

Remarque 1.2. *Le nombre de fois qu'une opération est appliquée est appelé l'ordre de différenciation (ou de désaisonnalisation). Daudin et. al. [1996]*

Une autre opération appelé rééchantillonage permet de créer une nouvelle série qui se prête mieux à la modélisation. Le rééchantillonnage implique la création de nouveaux échantillons à partir d'un ensemble de données existant (l'échantillon d'origine). Cette méthode permet de lisser les données afin de faire ressortir le comportement globale de la série et pour améliorer la précision des prédictions Arlot [2007], Brown [2004]. Il existe deux méthodes de rééchantillonnage :

- Le sur-échantillonnage : ce processus, connu sous le nom d'*oversampling* en anglais, consiste à augmenter le nombre d'observations de la classe minoritaire (une catégorie dans un ensemble de données qui a un nombre relativement faible d'observations par rapport à d'autres classes) jusqu'à atteindre un certain équilibre, ou du moins un niveau acceptable pour garantir la fiabilité des prédictions.
- Le sous-échantillonnage : en revanche, l'*undersampling* consiste à réduire le nombre d'observations de la classe majoritaire (une catégorie dans un ensemble de données qui a un nombre relativement élevé d'observations par rapport à d'autres classes) afin d'équilibrer le ratio avec la classe minoritaire.

1.3 Modèles de processus pour les séries chronologiques

Un modèle est une représentation simplifiée de la réalité qui cherche à exprimer les mécanismes opérant dans le phénomène étudié.

Les **modèles déterministes** se fondent sur des lois physiques, mathématiques ou théoriques, garantissant que des données d'entrée fixes génèrent systématiquement un résultat identique Prenat et. al. [2010]. On peut citer les modèles additifs et multiplicatifs, présentés ci-dessous.

Dans un modèle additif, la valeur observée d'une série chronologique est obtenue en additionnant les différentes composantes telles que la tendance, la saisonnalité et les variations résiduelles, sont indépendante les unes des autres. Un modèle additif est représenter par l'équation suivante :

$$Y_t = T_t + S_t + \varepsilon_t,$$

où :

- Y_t : est la valeur observée à un moment donné t ;
- T_t : représente la composante tendance à l'instant t ;
- S_t : représente la composante saisonnale à l'instant t ;
- ε_t : variations résiduelles ou bruit.

La Figure 1.2 représente une série temporelle avec des valeurs qui évoluent au fil du temps. La ligne bleue représente la série avec l'axe des abscisses représente les dates, l'axe des cordonnées représente la valeur de la série. Chaque point représente une observation de la série à une date spécifique.

Dans un modèle multiplicatif, les variations saisonnières ainsi que les variations résiduelles sont influencées par la tendance. Un modèle multiplicatif peut être représenté par l'équation suivante :

$$Y_t = T_t \times S_t \times \varepsilon_t,$$

La Figure 1.3 représente un modèle multiplicatif de série temporelle. La courbe bleue représente la série. Sur l'axe horizontal nous avons le temps et sur l'axe vertical les valeurs de la série. Cette série combine une tendance, une saisonnalité et un bruit.

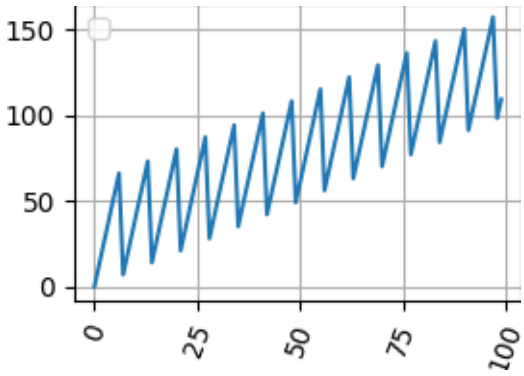

FIGURE 1.2 – Modèle Additif

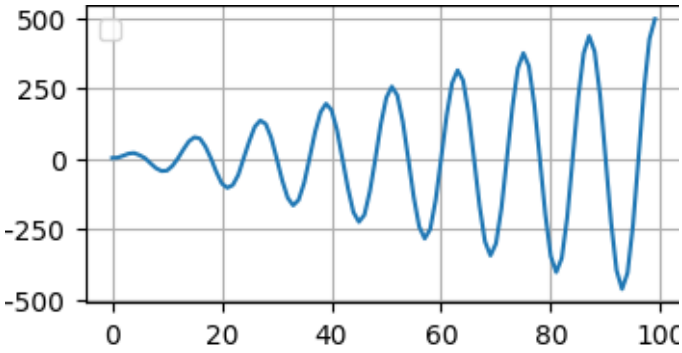

FIGURE 1.3 – Modèle Multiplicatif

A présent nous allons présenter les principaux modèle de processus aléatoire utilisés pour modéliser une série chronologique Moulines et Roueff [2010]. Les sous-sections 1.3.1 à 1.3.5 présentents successivement les modèles autoregressif (AR), de moyenne mobile (MA), les modèle combinés : ARMA, ARIMA, et SARIMA.

1.3.1 Processus autorégressif (AR)

Un processus autorégressif d'ordre $p \in \mathbb{N}^*$, noté par $AR(p)$ est un modèle de série temporelle où la valeur actuelle de la série est expliquée par une combinaison linéaire de ses valeurs passées plus un terme d'erreur.

Définition 1.7. *Un processus $(Y_t)_{t\in T}$ à valeurs réelles est autorégressif d'ordre p s'il vérifie l'équation suivante :*

$$Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + ... + \phi_p Y_{t-p} + \varepsilon_t, \tag{1.10}$$

où $\phi_1, \phi_2, ..., \phi_p \in \mathbb{R}$ sont les coefficients autorégressifs, avec $\phi_p \neq 0$ et ε_t est un terme d'erreur aléatoire ($\varepsilon_t \to \mathcal{N}(0, \sigma^2)$).

On peut réécrire le modèle, ce qui nous permet d'obtenir un modèle de la forme suivante :

$$\Phi(B)Y_t = \varepsilon_t, \tag{1.11}$$

où $\Phi(B)$ est le polynôme en B de degré p dont les coefficients sont $(1, -\phi_1, -\phi_2, -..., -\phi_p)$.
La fonction d'autocovariance est :

$$\gamma_h = \begin{cases} \phi_1\gamma_{h-1} + \phi_2\gamma_{h-2} + ... + \phi_p\gamma_{h-p} + \sigma_\varepsilon^2 & \text{, si } h = 0\,; \\ \phi_1\gamma_{h-1} + \phi_2\gamma_{h-2} + ... + \phi_p\gamma_{h-p} & \text{, si } h \geq 1. \end{cases} \tag{1.12}$$

La fonction d'autocorrélation :

$$\rho_h = \frac{\gamma_h}{\gamma_0} = \begin{cases} \phi_1\rho_{h-1} + \phi_2\rho_{h-2} + ... + \phi_p\rho_{h-p} + \frac{\sigma_\varepsilon^2}{\gamma_0} & \text{, si } h = 0\,; \\ \phi_1\rho_{h-1} + \phi_2\rho_{h-2} + ... + \phi_p\rho_{h-p} & \text{, si } h \geq 1. \end{cases} \tag{1.13}$$

Pour un processus $AR(p)$, **la fonction d'autocorrélation partielle** est nulle à partir de $p+1$.

• L'ACF d'un modèle AR(p) décroît progressivement, souvent de manière exponentielle ou sinusoïdale. Cette décroissance est due à la nature autorégressive du modèle, où chaque observation est une combinaison linéaire des observations précédentes. L'impact des observations passées diminue avec le temps, mais ne s'arrête pas brusquement.

• La PACF d'un modèle AR(p) présente une coupure nette après le lag p. Cela signifie qu'après ce point, les lags supplémentaires n'apportent pas d'information supplémentaire sur la série temporelle, car celle-ci est déjà capturée par les lags précédents.

Par exemple, le processus autorégressif AR d'ordre 1 (AR(1)) prend la forme suivante :

$$Y_t = \phi_1 Y_{t-1} + \varepsilon_t. \tag{1.14}$$

• $E(Y_t) = E(\phi_1 Y_{t-1} + \varepsilon_t) \Rightarrow E(Y_t) - \phi_1 E(Y_{t-1}) = 0$
si Y_t est stationnaire donc $E(Y_t) = E(Y_{t-1})$
• $var(Y_t) = \frac{\sigma^2}{1-\phi^2}$
Le paramètre ϕ détermine la stationnarité du processus $AR(1)$ tel que :

$$\phi = \begin{cases} < 1 & \text{, Le processus est stationnaire} \\ = 1 & \text{, Le processus non stationnaire (marche aléatoire)} \\ > 1 & \text{, Le processus est explosif} \end{cases}$$

Exemple 1.4. *Dans cet exemple, nous allons simuler un processus auto-régressif (AR) avec trois valeurs différentes du paramètre autorégressif et afficher les fonctions d'autocorrélation (ACF) et d'autocorrélation partielle (PACF).*

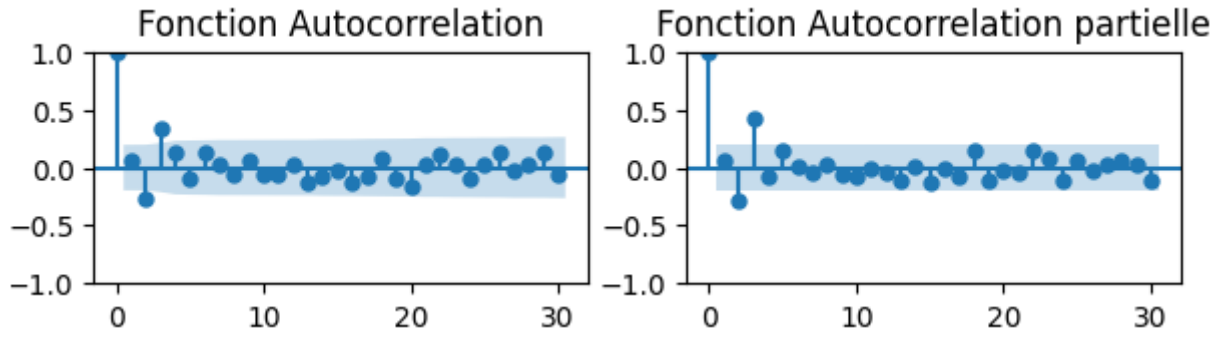

FIGURE 1.4 – Fonction d'autocorrélation et autocorrélation partielle du modèle AR(3) avec les trois paramètres $\phi_1 = 0.15$, $\phi_2 = -0.25$ et $\phi_3 = 0.5$

L'autocorrélation du processus AR(1) est donné par la formule suivante :

$$\rho_k = \frac{\sum_{i=1}^{p} \phi_i \gamma_{k-i}}{\gamma_0} \tag{1.15}$$

tel que ρ_k est l'autocorrélation au décalage k, γ_k est l'autocovariance au décalage k et γ_0 est la variance de la série.

1.3.2 Processus moyenne mobile (MA)

Dans un processus de moyenne mobile, chaque observation Y_t est générée par une somme pondérée de bruits blancs jusqu'à la q^{eme} période dans le passé.

Définition 1.8. *Un processus aléatoire $(Y_t)_{t\in T}$ est de de moyenne mobile d'ordre q $(q > 0)$ noté $MA(q)$ vérifie :*

$$Y_t = \varepsilon_t + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i}, \tag{1.16}$$

où $\theta_1, \theta_2, ..., \theta_q \in \mathbb{R}$ sont les coefficients de la moyenne mobile pour les différents retards, ε_t est un bruit blanc.

Un processus de moyenne mobile peut aussi être noté par :

$$Y_t = \Theta(B)\varepsilon_t, \tag{1.17}$$

où Θ est le polynôme de degré q dont les coefficients sont $(1, \theta_1, \theta_2, ..., \theta_q)$.

Par exemple, un processus $MA(1)$ prend la forme suivante :

$$Y_t = \theta_1 \varepsilon_{t-1} + \varepsilon_t, \tag{1.18}$$

avec $\varepsilon_t \to N(0, \sigma^2)$.
La fonction d'autocovariance est donnée par :

$$\gamma_h = \begin{cases} (1 + \theta_1^2 + \theta_2^2 + ... + \theta_q^2)\sigma_\varepsilon^2 & , \text{si } h = 0\,; \\ (\theta_h + \theta_1\theta_{h+1} + \theta_2\theta_{h+2} + ... + \theta_q\theta_{q-h})\sigma_\varepsilon^2 & , \text{si } 1 \le h \le q\,; \\ 0 & , \text{si } h > q. \end{cases} \tag{1.19}$$

La fonction d'autocorrélation :

$$\rho_h = \begin{cases} 1 & , \text{si } h = 0\,; \\ \frac{\theta_h + \theta_1\theta_{h+1} + ... + \theta_q\theta_{q-h}}{1 + \theta_1^2 + \theta_2^2 + ... + \theta_q^2} & , \text{si } 1 \le h \le q\,; \\ 0 & , \text{si } h > q. \end{cases} \tag{1.20}$$

Exemple 1.5. *On simule un modèle MA(2), et en affichant sa fonction d'autocorrélation (ACF) et sa fonction d'autocorrélation partielle (PACF).*

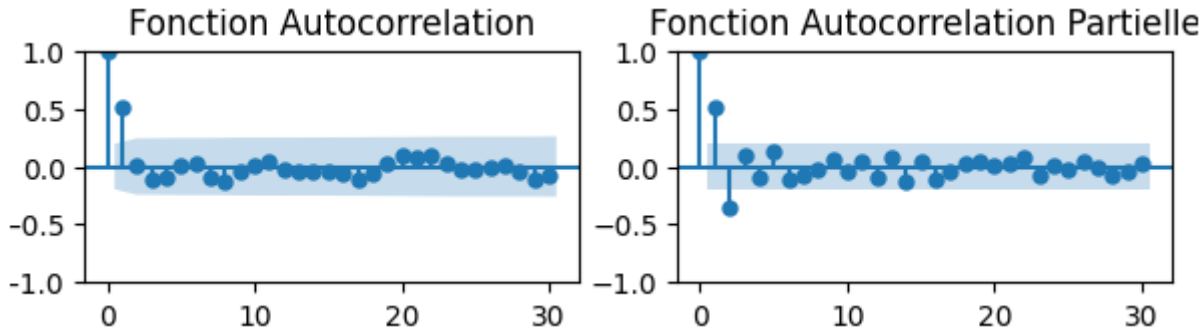

FIGURE 1.5 – Fonction d'autocorrélation et autocorrélation partielle du modèle MA(2) avec les deux paramètres $\theta_1 = 0.85$ et $\theta_2 = 0.25$

• Dans un modèle de moyenne mobile MA(q), l'autocorrélation (ACF) diminue brusquement après le lag q. Cette diminution est due au fait que l'effet des chocs passés (erreurs) est limité aux q périodes, et au-delà de ce seuil, la corrélation devient insignifiante.

• La fonction d'autocorrélation partielle (PACF) diminue progressivement, souvent de manière exponentielle ou sinusoïdale. Cette décroissance est due au fait que chaque terme de la série est une combinaison des erreurs passées, ce qui crée des dépendances plus complexes entre les observations.

1.3.3 Processus (ARMA)

Le modèle ARMA est la combinaison des deux modèles précédents en introduisant une dépendance du processus vis-à-vis de son passé modèle $AR(p)$ et un effet retardé des chocs modèle $MA(q)$. Un tel modèle, appelé autorégressif - moyenne mobile (ARMA), est caractérisé par le paramètre p de la partie autorégressive et le paramètre q de la partie moyenne mobile.

Définition 1.9. *Un processus autorégressif-moyenne mobile d'ordres p, q est la somme d'un autoregressif et d'un processus de moyenne-mobile. Il s'écrit sous la forme :*

$$Y_t = \sum_{i=1}^{p} \phi_i Y_{t-i} + \varepsilon_t + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} \tag{1.21}$$

Le modèle peut s'écrire comme suit :

$$\Phi(B) Y_t = \Theta(B) \varepsilon_t, \tag{1.22}$$

où Φ et Θ sont des polynômes de degrés respectifs p et q.
le modèle AR et le modèle MA sont deux cas particuliers du modèle ARMA :

- AR(p) = ARMA(p, 0).
- MA(q)= ARMA(0, q).

Les graphiques ci-dessus montrent la Fonction d'Autocorrélation (ACF) et la Fonction d'Autocorrélation Partielle (PACF) en simulant le modèle ARMA(2,3).

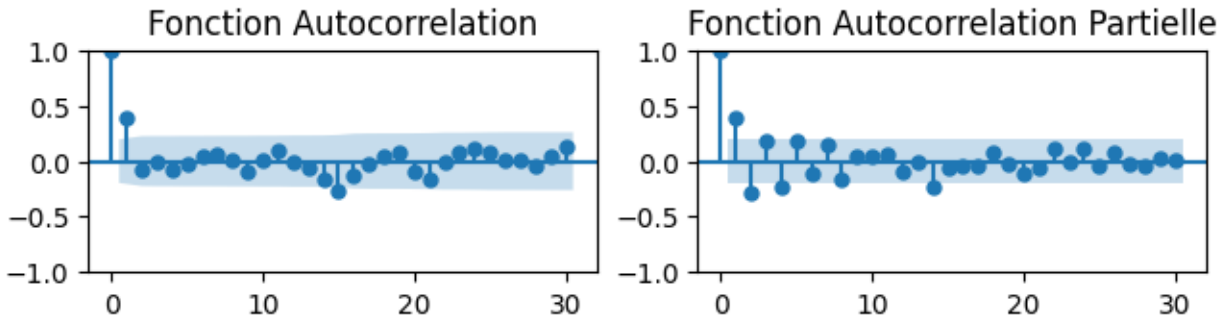

FIGURE 1.6 – Fonction d'autocorrélation et autocorrélation partielle du modèle ARMA(2,3) avec les paramètres $\phi_1 = 0.10$, $\phi_2 = 0.3$, $\theta_1 = 0.65$, $\theta_2 = -0.35$ et $\theta_3 = -0.15$

• L'ACF combine les caractéristiques des modèles autorégressifs (AR) et des modèles à moyenne mobile (MA). La décroissance de l'ACF dépend des valeurs spécifiques des paramètres p et q dans un modèle ARMA. Si p est grand, l'ACF décroît lentement, indiquant une forte dépendance aux valeurs passées. En revanche, si p est petit, l'ACF décroît plus rapidement.
• La PACF combine également les caractéristiques des modèles autorégressifs (AR) et des modèles à moyenne mobile (MA). Plus précisément, la PACF montre des coupures après p retards (lags) et des décroissances graduelles. Ces coupures indiquent les retards significatifs où l'influence directe d'une observation sur une autre est importante, sans tenir compte des autres observations. En d'autres termes, la PACF révèle la contribution spécifique de chaque retard à la corrélation.

1.3.4 Processus ARIMA

L'idée fondamentale derrière les modèles ARIMA est de traiter les séries temporelles non stationnaires avec une tendance. Pour ce faire, on applique une différenciation d'ordre d

suffisante au processus initial pour le rendre stationnaire, puis on applique un modèle ARMA à la partie différenciée :

$$Z_t = \Delta^d Y_t = (1-B)^d Y_t, \tag{1.23}$$

où Y_t représente la série originale, Z_t représente la série temporelle obtenue en différenciant Y_t et d indique le degré de différenciation appliqué.

L'équation d'un modèle ARIMA(p, d, q) peut être exprimée comme suit :

$$\Phi(B)\Delta^d Y_t = \Phi(B)Z_t = \Theta(B)\varepsilon_t, \tag{1.24}$$

où B est l'opérateur retard.

Exemple 1.6. *Dans cette exemple, on simule un modèle ARIMA(2,1,1) et analysons ses propriétés temporelles en affichant sa fonction d'autocorrélation (ACF) et sa fonction d'autocorrélation partielle (PACF).*

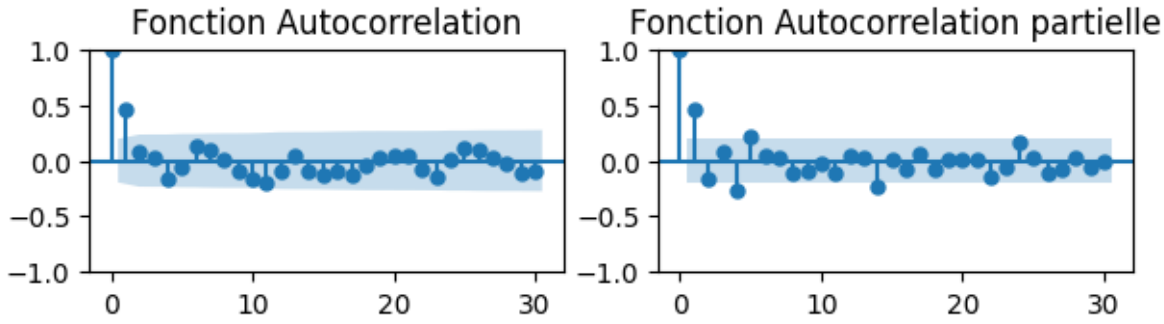

FIGURE 1.7 – Fonction d'autocorrélation et autocorrélation partielle du modèle ARIMA(2,1,1) avec les paramètres $\phi_1 = 0.95$, $\phi_2 = 0.05$, $\theta_1 = 0.65$ avec $d = 1$.

• L'acf ressemble à une série différenciée d'ordre d d'un ARMA(p, q). Peut présenter des comportements similaires à des ACF d'ARMA, mais après différenciation.

• Après différenciation, la PACF peut afficher des comportements analogues à ceux observés dans les PACF d'un modèle ARMA

1.3.5 Processus SARIMA

Les modèles SARIMA sont spécifiquement conçus pour modéliser des séries temporelles présentant des variations saisonnières. Ils combinent les composantes ARIMA avec des termes saisonniers pour capturer ces variations. Pour éliminer cette saisonnalité on applique **l'opérateur de désaisonnalité** .

Définition 1.10. *Un processus Y_t est considérée comme un processus SARIMA(p,d,q)(P,D,Q) avec une période s si la série différenciée $X_t = (1-B)^d(1-B^s)^D Y_t$ est stationnaire et si Y_t satisfait le modèle suivant :*

$$\Phi_p(B)\Phi_P(B^s)\Delta^d\Delta_s^D Y_t = \Theta_q(B)\Theta_Q(B^s)\varepsilon_t \tag{1.25}$$

où :

- *D ordre de différenciation saisonnière,*
- *P ordre de l'autorégression saisonnière,*
- *Q ordre de la moyenne mobile saisonnière,*
- *$\Phi_P(B^s) = 1 - \phi_1 B^s - ... - \phi_P B^{Ps}$ polynôme autorégressif saisonnier de degré P*
- *$\Theta_Q(B^s) = 1 + \theta_1 B^s + ... + \theta_1 B^{Qs}$ polynôme de moyenne mobile saisonnier de degré Q*

Exemple 1.7. *Danc cet exemple, nous simulons un processus SARIMA avec des coefficients spécifiques, puis analyser la série résultante à l'aide de graphiques de la Fonction d'Autocorrélation (ACF) et de la Fonction d'Autocorrélation Partielle (PACF).*

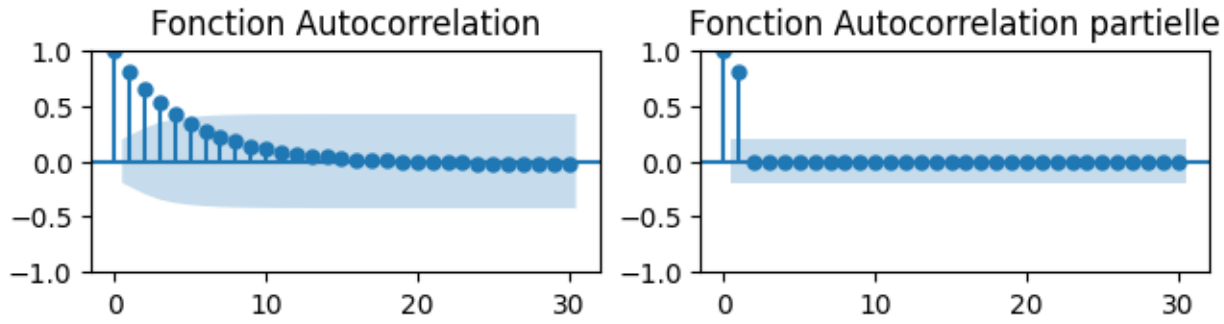

FIGURE 1.8 – Fonction d'autocorrélation et autocorrélation partielle du modèle SARIMA(2,1,2)(1,2,2) avec les paramètres $\phi_1 = 0.55$, $\phi_2 = 0.85$, $\theta_1 = -0.65$, $\theta_2 = -0.35$ avec $d = 1$, $D = 2$ et $s = 12$.

• L'ACF affiche des pics à des intervalles réguliers de s, avec des décroissances progressives entre ces pics.

• La PACF présente des pics à des intervalles réguliers de s, avec des décroissances progressives entre ces pics.

1.4 Simulation des séries chronologiques

Algorithm 1: Simulations du modèle AR(p)

Input:

1. p : ordre du processus AR ;
2. $\phi_i, i = 1, \ldots, p$: paramètres du processus ;
3. n : taille de l'échantillon ;
4. $(y_0, ..., y_{p-1})$: initialisation du processus.

Output: n prochaines réalisations $(y_p, y_{p+1}, \ldots, y_{n+p})$ du processus.

Générer n réalisations ϵ_t $N(0,1)$, pour $t = p, p+1, \ldots, n+p$;
for $t = p$ *à* $n + p$ **do**
  $y_t \leftarrow \phi^T Y_{[t-1:t-p]} + \epsilon_t$;
end

La génération d'un processus de moyennes mobiles ne dépends pas directement des valeurs passées du processus. Ainsi, il suffit de connaître les paramètres du modèle pour pouvoir générer une réalisation du processus.

Algorithm 2: Simulations du modèle MA(q)

Input:

1. q : ordre du processus MA ;
2. $\theta_j, j = 1, \ldots, q$: paramètres du processus ;
3. n : taille de l'échantillon ;

Output: n prochaines réalisations du processus.

Générer ϵ_t $N(0,1)$, pour t allant de 1 à q;
for $t = q+1$ *à* $n+q$ **do**
  Générer ϵ_t $N(0,1)$;
  $y_t \leftarrow \theta^T \epsilon_{[t-1:t-q]} + \epsilon_t$;
end

Algorithm 3: Simulations du modèle ARMA(p,q)

Input:

1. p, q : ordre du processus Arma ;
2. $\phi_i, i = 1, ..., p$, $\theta_j, j = 1, \ldots, q$: paramètres du processus ;
3. n : taille de l'échantillon ;
4. $(y_0, ..., y_{p-1})$: initialisation du processus.

Output: n prochaines réalisations du processus.

Poser $m \leftarrow \max(p, q) + 1$;
Poser $y_t \leftarrow 0, \forall t = p, \ldots, p+n$;
Poser $\epsilon_t \leftarrow \mathcal{N}_{0,1}$, $\forall t = 1, \ldots, q+n$ % générer n réalisations de bruit blanc;
for $t = m$ *à* $m+n$ **do**
  $y_t \leftarrow \phi^T Y_{[t-1:t-p]} + \theta^T \epsilon_{[t-1:t-q]} + \epsilon_t$;
end

Algorithm 4: Simulation du modèle ARIMA(p,d,q)

Input:

1. p, q : ordres du processus ARIMA ;
2. d : ordre de différenciation ;
3. $\phi_i, i = 1, ..., p$, $\theta_j, j = 1, \ldots, q$: paramètres du processus ARMA ;
4. n : taille de l'échantillon ;
5. $(y_0, ..., y_{\max(p,q)-1})$: initialisation du processus.

Output: n prochaines réalisations du processus.

Poser $m \leftarrow \max(p, q) + 1$;
Poser $y_t \leftarrow 0, \forall t = \max(p, q) + d, \ldots, \max(p, q) + d + n - 1$;
Poser $\epsilon_t \leftarrow \mathcal{N}_{0,1}$, $\forall t = 1, \ldots, \max(p, q) + d + n - 1$ % générer $\max(p, q) + d + n - 1$ réalisations de bruit blanc;
for $t = m$ *à* $m + n - 1$ **do**
  $y_t \leftarrow \phi^T Y_{[t-1:t-p]} + \theta^T \epsilon_{[t-1:t-q]} + \epsilon_t$;
  for $i = 1$ *à* d **do**
    $y_t \leftarrow y_t - \phi^T Y_{[t-i:t-i-p]}$
  end
end

Algorithm 5: Simulations du modèle SARIMA(p,d,q)(P,D,Q)s

Input:

1. p, q : ordre du processus Arma non saisonnier ;
2. d : ordre de différenciation non saisonnier ;
3. P, Q : ordre du processus Arma saisonnier ;
4. D : ordre de différenciation saisonnier ;
5. $\phi_i, i = 1, ..., p$, $\theta_j, j = 1, \ldots, q$: paramètres du processus ;
6. n : taille de l'échantillon ;
7. $(y_0, ..., y_{p-1})$: initialisation du processus.
8. $(y0, ..., yp - 1)$: initialisation du processus non saisonnier ;
9. $(Y0, ..., Ys - 1)$: initialisation des premières s valeurs du processus saisonnier.

Output: n prochaines réalisations du processus.

Poser $m \leftarrow \max(p, q, P, Q) + 1$;
Poser $y_t \leftarrow 0, \forall t = m, \ldots, m + n$;
Poser $\epsilon_t \leftarrow \mathcal{N}_{0,1}$, $\forall t = 1, \ldots, Q + n$ % générer n réalisations de bruit blanc;
for $t = m$ *à* $m + n$ **do**
  $y_t \leftarrow \phi^T Y_{[t-1:t-p]} + \phi^T Y_{[t-m:t-1]} + \theta^T \epsilon_{[t-1:t-q]} + \theta^T \epsilon_{[t-m:t-1]} + \epsilon_t$;
end

1.5 Conclusion

Dans ce premier chapitre, nous avons posé les bases théoriques nécessaires à la compréhension des séries chronologiques. Nous avons défini ce que sont les séries chronologiques et leurs applications, en mettant en évidence les propriétés essentielles. Ces concepts fondamentaux sont cruciaux pour l'analyse et la modélisation des données temporelles, et ils nous permettent de mieux appréhender les dynamiques sous-jacentes des séries temporelles avant d'appliquer des méthodes de clustering.

Nous avons également exploré diverses transformations de données. Ces transformations sont souvent nécessaires pour rendre les séries chronologiques stationnaires et pour isoler les composantes importantes comme les tendances et les saisons. Elles jouent un rôle crucial dans la préparation des données pour l'analyse et la modélisation.

Enfin, nous avons introduit les modèles fondamentaux utilisés pour l'analyse des séries chronologiques, tels que les modèles AR, MA, ARMA, ARIMA et SARIMA.

Chapitre 2

Clustering des séries chronologiques

2.1 Introduction

Dans ce chapitre, nous explorons la façon de diviser un ensemble de séries temporelles en groupes homogènes présentant des propriétés similaires, ainsi que la manière de classer une série temporelle dans l'un des clusters possibles. Ces procédures sont désignées comme des méthodes de classification. Lorsque la classification est réalisée sans référence à un ensemble connu de groupes similaires, elle est appelée méthodes de regroupement. Ce type de classification est également appelé classification non supervisée. En revanche, lorsque des séries sont déjà classées dans des groupes avec des étiquettes, et que l'objectif est de classer de nouvelles séries observées, on utilise des méthodes de discrimination. Cela correspond à de la classification supervisée ou de la reconnaissance de motifs supervisée. Dans le cadre du regroupement de séries temporelles, nous disposons d'un ensemble de K séries, et l'objectif est de les partitionner en groupes de sorte que chaque série soit classée dans un seul groupe, et que chaque groupe soit aussi homogène que possible. Le regroupement est une méthode essentielle en analyse de données, et ses applications sont nombreuses dans différents domaines scientifiques. Les approches de regroupement impliquent de choisir une mesure de proximité entre les séries, puis d'utiliser ces proximités pour former les groupes. Cela peut se faire en agglomérant les données en utilisant certaines proximités (comme avec les méthodes hiérarchiques), en partitionnant les données en fonction de leur similarité (comme avec K-means), en estimant un mélange de modèles de génération de données, ou en projetant les données dans un espace réduit pour trouver les clusters.

2.2 Qu'est-ce que le Clustering ?

Le clustering est une méthode de l'apprentissage machine, consistant à regrouper des ensembles de données en sous-ensembles plus ou moins homogènes en fonction de leur proximité ou de leur similitude Rokach et Maimon [2005]. Cette approche non supervisée est largement utilisée dans l'analyse de données, permettant ainsi l'application d'algorithmes de classification pour organiser les données individuelles en groupes distincts. Cette approche de classification est pertinente dans les situations où la collecte de données est complexe. Cependant, ce défi est fréquent car diverses mesures peuvent conduire à des regroupements distincts. Par conséquent, elle doit être choisie avec discernement en fonction des résultats recherchés et de la manipulation des données.

Avant d'appliquer des méthodes de clustering, il est indispensable d'effectuer un prétraitement des séries temporelles pour les préparer à l'analyse et pour garantir la qualité des résultats de clustering. Ce prétraitement comprend la normalisation ou la standardisation des séries pour rendre les échelles comparables, ainsi que le traitement des valeurs manquantes ou anormales qui pourraient altérer les résultats du clustering. De plus, des transformations telles que l'ajustement saisonnier ou la différenciation peuvent être réalisées pour diminuer

la dépendance temporelle et stabiliser la variance de la série.

La littérature relève différents type de Clustering de séries chronologiques, selon la nature des données, la méthodes utilisée, et l'objectif poursuivi. Le Clustering de séries chronologiques peut être partagé en trois catégories :

1. Clustering des données d'une seule série en sous-séquences. Chaque sous-séquence contient des données similaires formant une sous-suite de la série originale. Ce problème est appelé *segmentation de série chronologique* ;
2. Clustering des données de plusieurs séries en sous-séquences. Chaque sous-séquence contient des sous-suites des séries originales ;
3. Clustering d'un ensemble de séries chronologiques en plusieurs clusters regroupant des séries similaires.

Le clustering de séries chronologiques, et plus généralement de données se base sur la notion de *similarité*. Deux données similaires devraient être dans le même cluster. Il faudra donc définir la notion de similarité, afin de développer une méthode de Clustering.

Les sections suivantes sont dévolues à rapporter la revue de la littérature concernant la définition de fonction de similarité, puis les algorithmes de Clustering des séries chronologiques.

2.3 Applications du clustering de séries chronologiques

L'analyse de regroupement des séries chronologiques est une approche essentielle pour identifier des motifs et des structures cachées dans les données temporelles. Cette méthode trouve des applications variées dans divers domaines, notamment la détection d'anomalies, l'identification de tendances et la segmentation des comportements.

2.4 Mesures de similarité dans les séries chronologiques

Comme le clustering implique le regroupement d'instances ou d'objets similaires, il est nécessaire d'avoir une mesure permettant de déterminer la similitude ou la différence entre deux objets. Caiado et. al. [2006] et Daniel et Ruey S. [2021] rapportent les principales mesure de similarité entre deux séries. Zolhavarieh et. al. [2014] rapportent les mesures de similarité utilisées pour le clustering de séries en sous-séquences.

Définition 2.1. *Une fonction de dissimilarité (Peña et Tsay [2021]) est une fonction réelle* $d : X \times X \longrightarrow \mathbb{R}$ *positive et symétrique, définie entre chaque paire de série* x *et* y *de l'ensemble des séries* X*, telle que :*
• $d(x, y) = 0 \Longleftrightarrow x = ay + b$*, où* a, b *sont des constantes réelles.*

Remarques 2.1.

1. *On a* $0 \leq d(x, y) \leq 1$*, où une dissimilarité nulle signifie que les deux séries sont absolument similaires, et égale à 1 si elle sont complètement différentes ;*
2. *la distance est un cas particulier de la fonction de dissimilarité, puisque toute distance possède les propriétés précédentes ainsi que l'inégalité triangulaire :* $d(x, y) \leq d(x, z) + d(z, y)$ *pour* $x, y, z \in X$ *;*
3. *dans le cas d'une série normalisée, on a* $d(x, y) = 0 \Longleftrightarrow x = y$ *;*
4. *on peut associer une* ***fonction similarité*** *à* d*, de la manière suivante :* $s(x, y) = M - d(x, y)$ *; où* M *est la plus grande dissimilarité dans le jeu de données ;*

Contrairement aux fonctions de dissimilarité, où des valeurs plus élevées indiquent une plus grande dissimilarité, des valeurs plus élevées dans une fonction de similarité indiquent une similitude accrue entre les objets.

Catégorie	Application du clustering
Aviation /Astronomie	-Données astronomiques -prétraitement pour la détection des valeurs aberrantes
Biologie	-Clustering fonctionnel des données de séries chronologiques d'expression génique -Identification des gènes fonctionnellement liés
Climat	-Découverte d'indices climatiques
Énergie	-Découverte des schémas de consommation d'énergie
Environnement et urbanisme	-Analyse de la variabilité régionale des extrêmes du niveau de la mer
Finance	-Recherche de schémas saisonniers -Schéma de revenu du personnel -Création d'un portefeuille efficace -Découverte de schéma à partir des séries chronos boursières -Réduction des risques des portefeuilles
Médecine	-Détection de l'activité cérébrale
Psychologie	-Analyse du comportement humain dans le domaine psychologique
Robotique	-Formation de représentations prototypiques des expériences du robot
Reconnaissance vocale	-Vérification du locuteur -Classification biométrique de la voix à l'aide du clustering hiérarchique

TABLEAU 2.1 – Applications du Clustering de séries chronologiques. Tiré de Aghabozorgi et. al. [2015].

Définition 2.2. *Soit $x, y \in \mathbb{R}^p$. La distance de Minkowski est donné pour $m > 0$, comme suit :*

$$D_m(x, y) = \left(\sum_{i=1}^{p} |x_i - y_i|^m \right)^{\frac{1}{m}}. \tag{2.1}$$

Remarques 2.2.

- *Lorsque $m = 1$, cette mesure correspond à la distance de Manhattan ;*
- *lorsque $m = 2$, elle équivaut à la distance euclidienne ;*
- *une propriété intéressante de cette famille de distance est la suivante :*

$$(m > k \geq 1) \Longrightarrow (D_m(\Theta_x, \Theta_y) \leq D_k(\Theta_x, \Theta_y)).$$

L'inconvénient de cette distance est qu'elle n'est significative que si les valeurs x_i et y_i sont alignées, autrement dit si les séries $(x_t)_{t\in T}$ et $(y_t)_{t\in T}$ sont synchronisées. La distance suivante compare la pente de deux séries en chaque instant. La pente (à droite) à l'instant k de la série $(x_t)_{t\in T}$ est donnée par le rapport :

$$\frac{x_{k+1} - x_k}{t_{k+1} - t_k}.$$

Définition 2.3. *soient $x, y \in \mathbb{R}^n$. La distance STS (Short Time Series) est donnée par :*

$$d_{STS}(x, y) = \left(\sum_{i=1}^{n} \left(\frac{x_{k+1} - x_k}{t_{k+1} - t_k} - \frac{y_{k+1} - y_k}{t_{k+1} - t_k} \right)^2 \right)^{1/2}. \tag{2.2}$$

Zolhavarieh et. al. [2014] recommandent de normaliser les données pour avoir les mêmes ordre d'échelles.

La Dynamic Time Warping (DTW) est une méthode mathématique pour évaluer la similarité entre deux séries temporelles qui ne s'alignent pas parfaitement. Cette approche est largement employée dans l'analyse de données pour calculer la distance entre des séries temporelles (Senin [2008]).

Définition 2.4. *La distance DTW des séquences x, y est exprimée sous la forme du problème d'optimisation suivant :*

$$D_{i,j} = \begin{cases} |x_1 - y_1| & , \text{ si } i = j = 1 \text{ ;} \\ |x_i - y_j| + min(D(i-1,j), D(i-1,j-1), D(i,j-1)) & , \text{ sinon.} \end{cases} \tag{2.3}$$

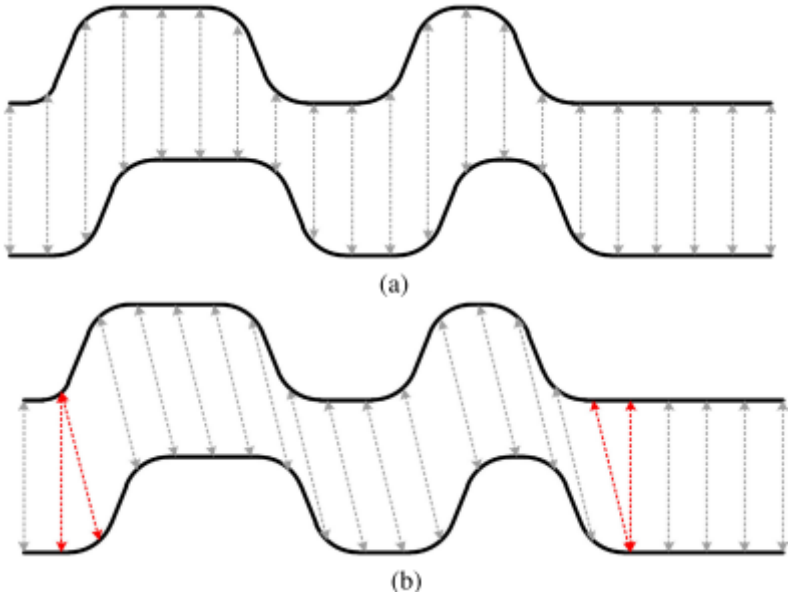

FIGURE 2.1 – Alignement de deux séries temporelles (les flèches représentent l'alignement points). (a) Distance euclidienne, (b) Distance DTW.

Comme le montre la Figure 2.1, la distance euclidienne mesure la similarité exactement à la même position sur l'axe du temps, par contre la DTW ne compare pas forcément deux valeurs des séries prises au même temps. En effet, les lignes joignant les deux séries représentent la comparaison retenue par la DTW. On remarque que sur cet exemple, la DTW arrive à détecter le décalage. Signalons que les distance et mesure de similarité précédentes ont été utilisées avec ou sans modèle d'ajustement (AR, MA, ...) pour la détection de valeurs et séquences de valeurs aberrantes.

Une autre façon de mesurer la distance entre deux séries est de représenter la série $(x_t)_{t\in T}$ par un vecteur $\Theta_x = (y_1, y_2, \ldots, y_p)$ (voir Peña et Tsay [2021]). Par exemple, on peut représenter une série par ses statistiques : min, max, moyenne, quartiles. Puis d'adopter une distance de l'espace $\mathbb{R}^p$ sur les vecteurs Θ_x, Θ_y. Une possible distance est celle de Minkowski.

Une autre façon de mesurer la distance en utilisant l'approche précédente est de prendre les h premiers termes de la fonction d'autocorrelation (ACF). D'autre distances du même type peuvent être obtenues, en prenant les fonctions d'autocorrelations partielle ou les autocorrelations inverses. La généralisation apportée à cette classe de distance consiste à intégrer une matrice de poids relatifs aux autocorrelations. La définition suivante généralise cette distance.

Définition 2.5. *Soient x, y deux série chronologique.*

$$d_M(x,y) = \sqrt{(\rho_x - \rho_y)^T M (\rho_x - \rho_y)}, \tag{2.4}$$

où :

1. *ρ_x : vecteur des autocorrélations, autocorrélations partielles, ou des autocorrélations partielles inverse ;*
2. *M : est une matrice de poids (symétrique et définie positive).*

Quand ρ_x est le vecteur des autocorrélations et M est la matrice identité, on obtient la distance euclidienne entre les autocorrélations. Quand c'est la matrice inverse des variances-covariances des autocorrélations, on obtient la distance de Mahalanobis entre les autocorrélations.

Caiado et. al. [2006] proposent d'utiliser le periodogramme comme représentation de la série, puis d'appliquer la distance euclidienne. Les auteurs introduisent aussi le periodogramme normalisé par la variance empirique de la série, ainsi que le logarithme du periodogramme. Une distance entre x et y peut être définie par :

$$d_P(x,y) = \sqrt{\sum_{j=1}^{\lfloor \frac{n}{2} \rfloor} \left(P_x(w_j) - P_y(w_j)\right)^2} \tag{2.5}$$

Si nous ne sommes pas intéressés par l'échelle du processus mais seulement par sa structure de corrélation, il est préférable d'utiliser le périodogramme normalisé (ou périodogramme rescalé) en remplaçant P_{w_j} dans cette mesure par $NP_{w_j} = \frac{P_{w_j}}{\hat{\sigma}_0^2}$, où $\hat{\sigma}_0^2$ est la variance échantillonnale de la série chronologique. Ainsi,

$$d_{NP}(x,y) = \sqrt{\sum_{j=1}^{\lfloor \frac{n}{2} \rfloor} \left(NP_x(w_j) - NP_y(w_j)\right)^2}. \tag{2.6}$$

Puisque la variance des ordonnées du périodogramme est proportionnelle à la valeur du spectre aux fréquences correspondantes, il est pertinent d'utiliser le logarithme du périodogramme normalisé,

$$d_{LNP}(x,y) = \sqrt{\sum_{j=1}^{\lfloor \frac{n}{2} \rfloor} \left(\log NP_x(w_j) - \log NP_y(w_j)\right)^2}. \tag{2.7}$$

Comme on peut s'y attendre, les mesures basées sur les autocorrélations et celles basées sur le périodogramme sont liées. Il est bien connu que le périodogramme a la représentation équivalente $P_{w_j} = 2\hat{\sigma}_0^2 + \frac{2}{n}\sum_{k=1}^{n-1} \hat{\rho}_k \cos(w_j k)$,où $\hat{\rho}_k$ est la fonction d'autocovariance échantillonnale (pour plus de détails, voir Brockwell et Davis [1991]). En divisant P_{w_j} par $\hat{\sigma}_0^2$, on obtient le périodogramme normalisé donné par

$$NP(w_j) = 2\left[1 + 2\sum_{k=1}^{n-1} \hat{\rho}_k \cos(w_j k)\right] \tag{2.8}$$

Une autre façon de définir une distance est de représenter une série par ses paramètre autoregressif. Cela consiste à ajuster un modèle autoregressif $AR(p)$, puis de représenter la série x par ses paramètres. Il est importants à cette étape de considérer aussi les paramètres nuls. L'exemple suivant clarifie cette approche :

Exemple 2.1. *Considérons deux séries $(x_t)_{t\in T}$, $(y_t)_{t\in T}$ issues d'un processus autoregressif $AR(2)$. La première a les paramètres $\phi_1 = 0.2$, $\phi_2 = 0.6$. La deuxième a les paramètres $\phi_1 = 0.3$ et $\phi_4 = 0.8$. La représentation donnera : $\Theta_x = (0.2, 0.6, 0, 0)$ et $\Theta_y = (0.3, 0, 0, 0.8)$. La distance euclidienne donnera $D_2(\Theta_x, \Theta_y) \approx 1.0049$.*

Peña et Tsay [2021] recommandent de prendre p assez grand pour décrire convenablement la dépendance linéaire existant dans les séries. Deuxièmement, regrouper les paramètres MA et AR d'une série ARIMA ne doit pas être fait, car des séries avec des paramètres différents peuvent être très similaires.

2.4.1 Visualisation graphiques de quelques distances de séries chronologiques

Dans cette section, nous évaluons la qualité des clusters produit avec les distances euclidienne, DTW, ACF, et PACF. On veut savoir si les clusters sont de formes sphériques, et s'ils sont bien séparé les uns des autres. Pour celà, on a choisi de visualiser les séries chronologiques dans le plan, en nous basons sur la matrice des distances produite par ces métriques. Pour ce faire, nous avons utilisé la méthode MDS (multidimensional scaling). Cette méthode prend une matrice de distances ou de similarité, et trouve des coordonnées dans $\mathbb{R}^k$, de tel manière que la matrice des distances de ces points soit très proche de la matrice des distances fournie.

La procédure qu'on a suivi est la suivante :

1. prendre 3 processus de série chronologiques ;
2. générer 10 séries chronologiques issues pour chaque processus ;
3. calculer les 4 matrices des distances correspondantes au 4 métriques : euclidienne, DTW, ACF, PACF ;
4. appliquer la méthode MDS, pour chaque matrice des distances, pour obtenir les coordonnées dans $\mathbb{R}$š ;
5. visualiser les 30 séries sur le plan.

La figure 2.2 compare des des processus $AR(1)$ avec les valeurs $\phi = (0.1), (0.5), (0.9)$. Les graphiques montrent que les points suivants :

- le cluster de la série $AR(1)$ avec $\phi = 0.9$ est beaucoup plus large que les autres ;
- les clusters sont plutôt bien séparés avec toutes les métriques ;
- l'ACF et PACF réussissent mieux à séparer les clusters ;
- la forme des clusters est plutôt la même.

La figure 2.3 compare des processus $MA(2)$ avec les paramètres $\theta : (0.1, 0.9), (0.5, 0.5) et (0.9, -0.3)$. On remarque les points suivants :

- les distances euclidienne et DTW échouent à séparer les clusters ;
- l'ACF et PACF donnent une meilleure séparation, en particulier la PACF ;
- les clusters donnés par l'ACF et PACF sont de formes convexes et de même tailles.

La Figure 2.4 compare des processus $ARMA$ de paramètres $\phi, \theta : ((0.1), (0.4, 0.3)), ((0.7), (-0.2, -0.6)$ On remarque les points suivants :

- les distances euclidienne et DTW échouent à séparer les cluster, mais réussissent mieux qu'avec les processus MA ;
- la distance ACf mélange le premier et le troisième cluster, mais sépare convenablement le deuxième cluster ;
- la distance PACF réussi bien à séparer ces clusters.

En conclusion, on remarque que la distance PACF est celle qui réussi à chacun des tests que l'on a effectué. L'échec des autres distances se situe au niveau des processus MA. En particulier lorsque ces un processus MA pure.

2.5 Méthodes de clustering

Parmi les diverses méthodes Ahmed et. al. [2020] de clustering disponibles, on trouve :

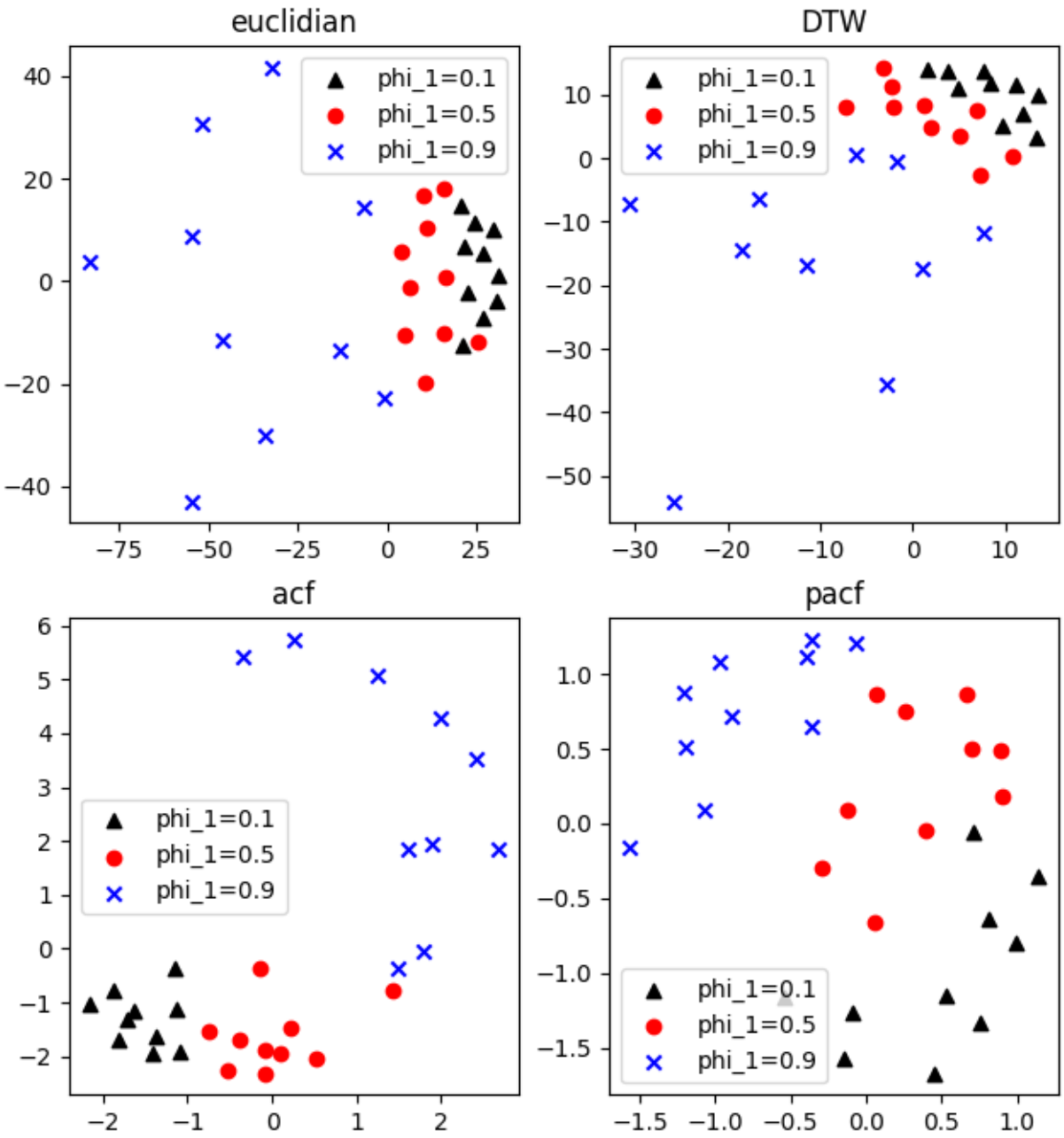

FIGURE 2.2 – Comparaison des mesures de distance et fonctions d'autocorrélation pour des processus AR(1)

2.5.1 Clustering Partitionnel

Un clustering partitionnel (Hammouda [2009]) consiste à séparer l'ensemble des données en sous-groupes distincts (clusters) qui ne se superposent pas, de manière à ce que chaque donnée soit assignée à un seul et unique sous-groupe. Le principe fondamental de ce clustering consiste à commencer avec un seul cluster, puis à le partitionner de manière itérative en redistribuant les objets ou en identifiant les clusters comme des régions densément peuplées jusqu'à ce qu'un critère d'arrêt soit rencontré. Le but de ces méthodes est de diviser de manière optimale l'ensemble des objets en un nombre prédéfini de groupes. Les clusters ainsi identifiés tendent à avoir une forme sphérique. Parmi les algorithmes connus on trouve le k-means.

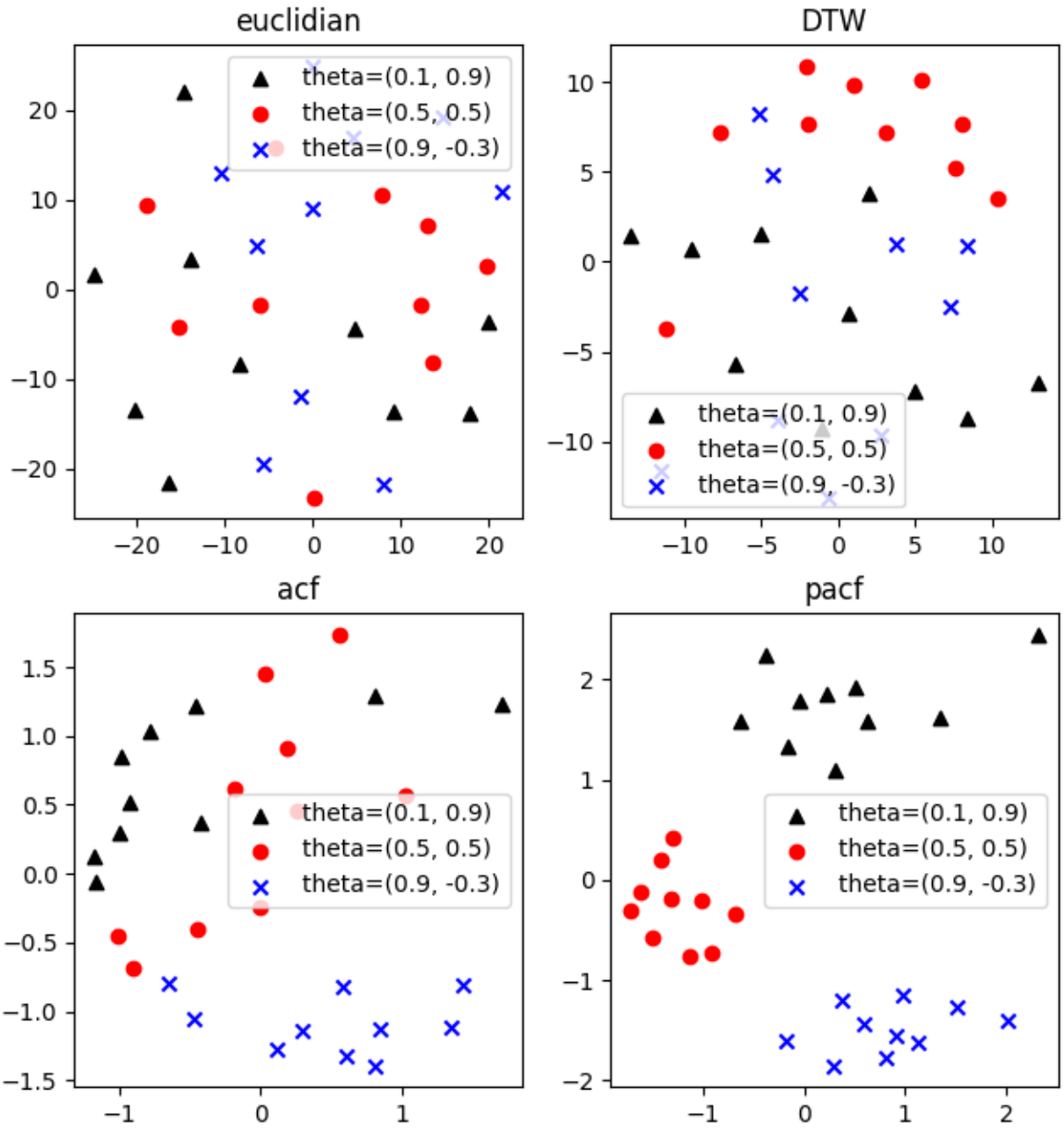

FIGURE 2.3 – Comparaison des mesures de distance et fonctions d'autocorrélation pour des processus MA(2)

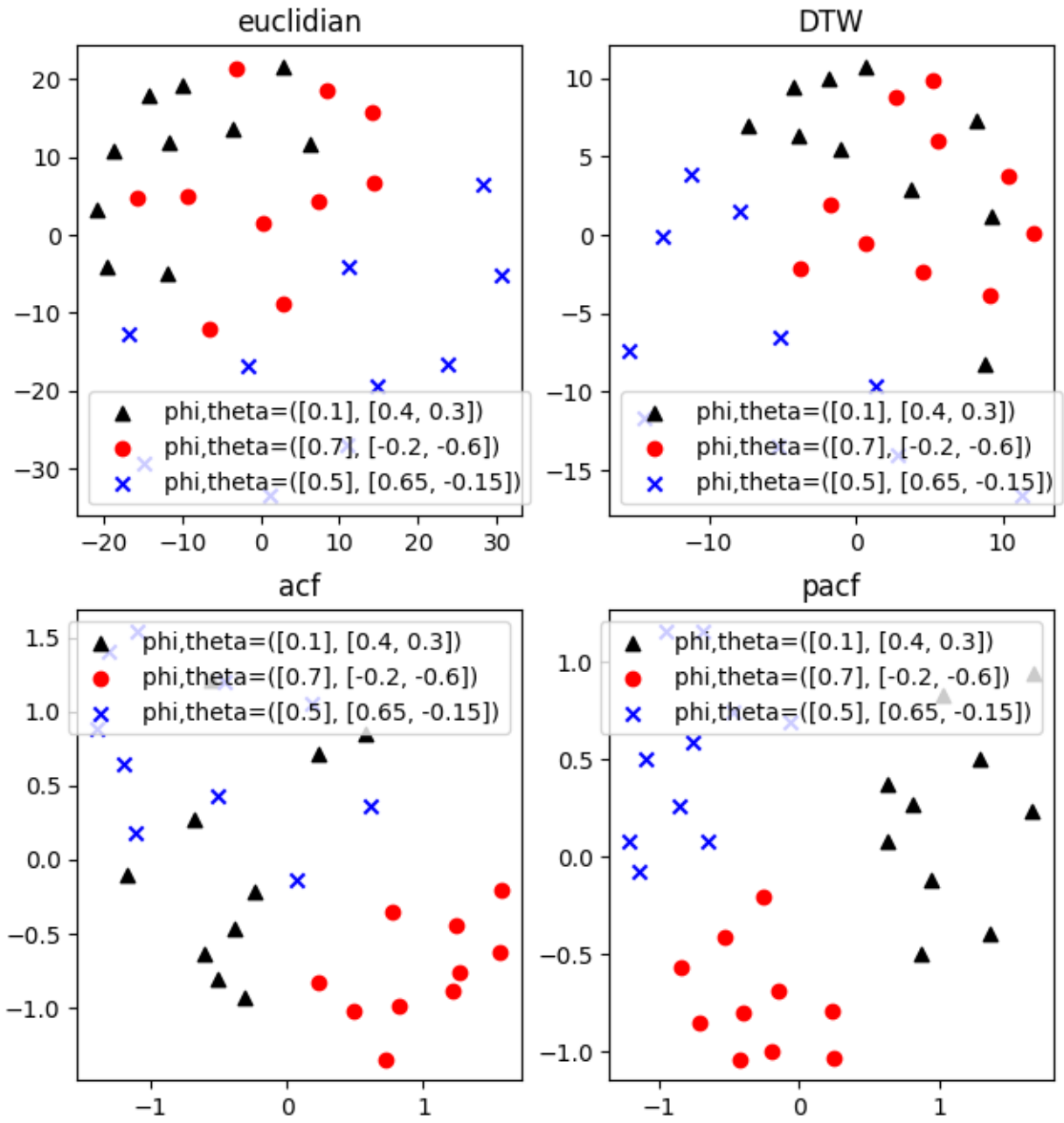

FIGURE 2.4 – Comparaison des mesures de distance et fonctions d'autocorrélation pour des processus ARMA(1,2)

2.5.1.1 L'algorithme K-means

L'algorithme K-means est un algorithme de partitionnement non supervisé souvent utilisé pour l'exploration de données et la reconnaissance de formes. Son objectif est de diviser un ensemble de K objets (ou K séries temporelles) caractérisés par p variables en un nombre donné de groupes G, de manière à ce que chaque groupe soit aussi homogène que possible. Chaque vecteur x_i représentant un objet contient les composantes x_{ij} correspondant aux valeurs des variables j. Les groupes doivent être mutuellement exclusifs et chaque objet doit appartenir à un seul groupe.

Étapes de l'algorithme K-means :

1. Choisir K points comme centroïdes des groupes initiaux. Cela peut être fait en :
   (a) assignant aléatoirement des objets aux K groupes et en prenant les centroïdes des groupes ;

(b) prenant comme centroïdes les K points les plus éloignés les uns des autres ;
(c) construisant des groupes initiaux en utilisant des informations préalables et en calculant leurs centroïdes, ou en sélectionnant leurs centroïdes a priori.

2. Calculer les distances euclidiennes de chaque objet aux K centroïdes, et assigner un objet au groupe avec la distance la plus courte. Une fois qu'un objet est réassigné, les centroïdes des groupes affectés sont recalculés.
3. Vérifier si la réassignation de certains objets améliore le critère SSW. Cela se produit si un objet dans un groupe est plus proche du centroïde d'un autre groupe. Dans ce cas, il est déplacé vers le nouveau groupe et les centroïdes des deux groupes affectés par le changement sont recalculés.
4. Lorsque le critère ne peut plus être amélioré, arrêter l'algorithme.

Le résultat de l'algorithme K-means peut dépendre de l'affectation initiale des points et de l'ordre des éléments. Il est donc recommandé de répéter l'algorithme avec différentes valeurs de départ et en permutant les éléments de l'échantillon pour s'assurer que l'ordre n'a pas d'influence significative.

L'idée principale de K-means est de minimiser la variation intra-groupe, mesurée par la somme des carrés intra-groupes (SSW), pour obtenir des groupes homogènes. La formule pour SSW est :

$$\text{SSW} = \sum_{g=1}^{G} \sum_{j=1}^{p} \sum_{i=1}^{n_g} (x_{ijg} - \bar{x}_{jg})^2 \tag{2.9}$$

où x_{ijg} est la valeur de la variable j pour l'objet i dans le groupe g, et $\bar{x}_{jg}$ est la moyenne de la variable j dans le groupe g.

Une autre manière de formuler le critère de minimisation est :

$$\min \text{SSW} = \min \sum_{g=1}^{G} \sum_{j=1}^{p} n_g s_{jg}^2, \tag{2.10}$$

où s_{jg}^2 est la variance de la variable j dans le groupe g.

Un autre critère consisterait à minimiser les distances euclidiennes au carré entre les objets et leurs centroïdes de groupe :

$$\min \sum_{g=1}^{G} \sum_{i=1}^{n_g} (x_{ig} - \bar{x}_g)'(x_{ig} - \bar{x}_g) = \min \sum_{g=1}^{G} \sum_{i=1}^{n_g} d^2(i, g) \tag{2.11}$$

Les critères SSW et de trace utilisés par K-means ont deux propriétés importantes. Premièrement, ils ne sont pas invariants aux changements d'échelle. Deuxièmement, minimiser la distance euclidienne tend à produire des groupes de forme sphérique. Concernant l'échelle, si les variables sont exprimées dans des unités différentes, il est préférable de les standardiser pour que le résultat du K-means ne dépende pas des variations d'échelle non pertinentes. Cependant, lorsque les variables sont dans les mêmes unités, il est souvent préférable de ne pas standardiser, car une variance plus élevée dans une variable peut indiquer la présence de plusieurs groupes distincts, et cette information pourrait être perdue si les données sont standardisées.

Les groupes sphériques produits par K-means peuvent poser problème lorsque les groupes ont des formes très différentes. Dans ce cas, des transformations des variables peuvent être appliquées pour que leur distribution conjointe au sein des groupes soit approximativement normale avec une matrice de covariance diagonale.

2.5.1.2 DBSCAN

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) est un algorithme de clustering non supervisé basé sur la densité couramment utilisé pour détecter des clusters denses dans des ensembles de données. Bien qu'il ait été initialement conçu pour des données spatiales, il peut également être adapté aux séries temporelles.

Voici comment fonctionne DBSCAN :

1. **Principe de base** :
   • DBSCAN se base sur la densité des points dans l'espace. Il identifie les régions denses en recherchant des zones où la densité des points est supérieure à un seuil donné.
   • Il ne nécessite pas de spécifier le nombre de clusters à l'avance, ce qui le rend plus flexible que d'autres algorithmes de clustering.
2. **Paramètres :**
   • **eps** : C'est la distance maximale entre deux points pour qu'ils soient considérés comme voisins. Si la distance entre deux points est inférieure à ϵ, ils sont regroupés.
   • **MinPts** : C'est le nombre minimum de points requis pour former un cluster. Si un groupe de points a au moins MinPts voisins, il est considéré comme un cluster.
3. **Points principaux et points frontière** :
   • Un point principal est un point qui a au moins **minPts** voisins dans un rayon **eps**.
   • Un point frontière est un point qui n'est pas un point principal mais qui est dans le voisinage d'un point principal.
4. **Étapes de l'algorithme** :
   • **Sélection du point initial** : choisir un point non visité et marquez-le comme visité.
   • **Vérification des voisins** : si le nombre de voisins de ce point (dans un rayon eps) est supérieur ou égal à minPts, un nouveau cluster est créé.
   • Ce point devient un point principal et ses voisins directs sont ajoutés au cluster.
   • Le processus est répété pour chaque voisin, en ajoutant leurs voisins au cluster s'ils sont aussi des points principaux.
   • Si un point visité n'a pas suffisamment de voisins (minPts), il est marqué comme bruit.
   • Continuez jusqu'à ce que tous les points soient visités.
5. **Avantages et Limitations** :
   Avantages :
   • DBSCAN peut détecter des clusters de formes complexes, ce qui le rend utile pour des données temporelles avec des motifs variés.
   • DBSCAN est moins sensible aux valeurs aberrantes que certaines autres méthodes de clustering, car il se base sur la densité locale plutôt que sur la distance euclidienne.
   • Pas besoin de spécifier le nombre de clusters à l'avance.
   Limitations :
   • Sensibilité aux paramètres eps et minPts.
   • Peut avoir des difficultés avec des densités de clusters variables.
   • Calcul coûteux de la matrice de distance

2.5.2 Clustering hiérarchique

La classification hiérarchique, consiste à effectuer une suite de regroupements en classes de moins en moins fines en agrégeant à chaque étape les objets ou les groupes d'objets les plus proches. Le nombre d'objets n'est pas prédéterminé à l'avance, mais sera déterminé après l'analyse. Cette approche fournit plusieurs partitions de l'ensemble des objets. On distingue deux types de méthodes : les méthodes ascendantes (ou algorithmes agglomératifs) et les méthodes descendantes (ou algorithmes divisifs). Ghosal et. al. [2020]

2.5.2.1 Méthode ascendante / Descendante

Ces approches sont parmi les plus anciennes et les plus couramment utilisées en classification automatique. Imaginons que nous ayons N objets à classifier, les algorithmes agglomératifs suivant cette approche commencent par une partition initiale où chaque objet est dans sa propre classe. Ensuite, ils fusionnent progressivement les classes jusqu'à ce que tous les objets appartiennent à une seule classe. À chaque fusion, il est nécessaire de recalculer les dissimilarités entre les nouvelles classes. Le choix des classes à fusionner est guidé par un critère spécifique à la méthode utilisée.

Algorithme agglomératif

- **Etape 1** : Déterminer toutes les dissimilarités inter-objets.
- **Etape 2** : Construire une classe à partir des deux plus proches objets ou classes.
- **Etape 3** : Redéfinir les dissimilarités entre la nouvelle classe et les autres objets ou classes (toutes les autres dissimilarités ne changent pas).
- **Etape 4** : Retour à l'étape 2 jusqu'à ce que tous les objets soient dans la même classe.

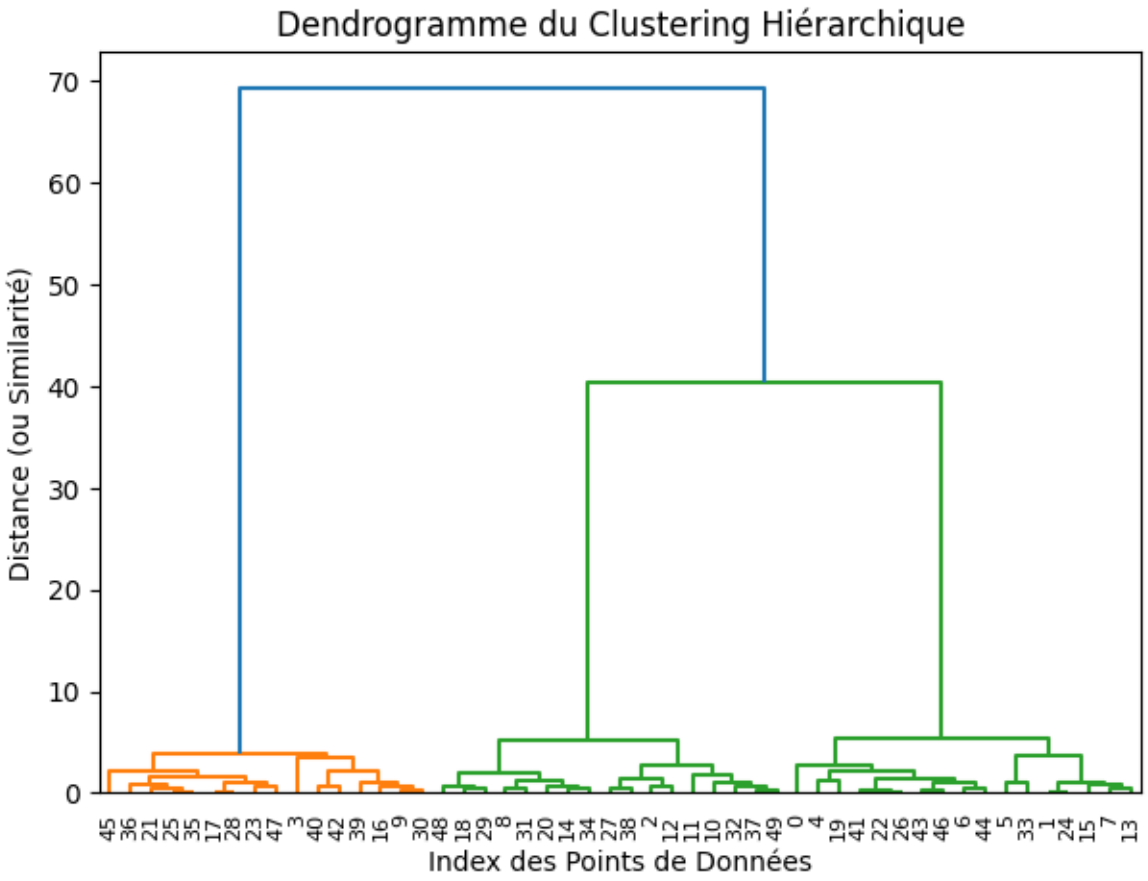

FIGURE 2.5 – Méthode ascendante

La Figure 2.5 montre le dendrogramme illustre les résultats du clustering hiérarchique agglomératif appliqué à nos données synthétiques. Dans cette représentation, chaque point de données est disposé le long de l'axe des x en fonction de son index, avec 50 points affichés dans notre cas. L'axe des y quantifie la distance entre les clusters, où une valeur plus élevée indique une dissimilarité accrue entre les clusters reliés. Les branches verticales dépeignent les clusters formés à chaque étape de l'algorithme, avec une hauteur plus grande signifiant une différence plus marquée entre les clusters. Les branches horizontales représentent les clusters eux-mêmes, sans signification particulière dans leur longueur. En traçant une ligne horizontale à travers les branches les plus longues du dendrogramme (sans couper de branches

verticales), vous pouvez déterminer le nombre optimal de clusters, identifié par le nombre de branches verticales traversées par la ligne.

Dans le paragraphe précédent, nous avons observé que la classification hiérarchique ascendante cherche à optimiser un seul critère à la fois, ce qui conduit à une séparation optimale. Cependant, cela peut entraîner des problèmes tels que l'effet de chaînage (deux entités très dissimilaires aux extrémités d'une longue chaîne peuvent appartenir à la même classe) ou l'effet de dissection (deux entités très similaires peuvent être dans des classes différentes). Pour résoudre ces problèmes, les algorithmes de classification hiérarchique descendante sont utilisés. Bien que moins populaires que les méthodes ascendantes, les algorithmes descendantes commencent par former une seule classe qui inclut tous les objets. Ensuite, ils choisissent une classe de la partition en cours en utilisant un premier critère local, puis ils procèdent à une bipartition successive des classes choisies en utilisant un deuxième critère local. Ce processus de bipartition se poursuit jusqu'à ce que tous les objets soient assignés à des classes différentes.

algorithme divisif

- **Etape 1** : Déterminer toutes les dissimilarités inter-objets.
- **Etape 2** : Choisir selon un critère local une classe.
- **Etape 3** : Partitionner la classe choisie en deux classes suivant un deuxième critère local.
- **Etape 4** : Redéfinir les dissimilarités entre la nouvelle partition et les autres classes.
- **Etape 5** : Retour à l'étape 2 jusqu'à ce que chaque objet soit dans une seule classe.

2.5.3 Qualité d'un Clustering par le coefficient silhouette et l'ars

Pour évaluer la qualité des résultats des méthodes de Clustering, un test d'homogénéité est nécessaire. Ce test est réalisé après avoir atteint une convergence où le résultat final du regroupement est identique au regroupement précédent, c'est-à-dire qu'aucune donnée ne change de cluster. Le test utilise le coefficient de silhouette. Le calcul du coefficient de silhouette commence par la recherche de la distance moyenne entre le i-ème point de données et toutes les données du même cluster, en supposant que ce point se trouve dans le cluster A. La formule de a_i est donnée comme suit :

$$a_i = \frac{1}{|A| - 1} \sum_{j \in A, j \neq i} d(i, j) \tag{2.12}$$

où :

A = nombre de données dans le cluster A

Ensuite, nous calculons la valeur b_i, qui est la valeur minimale de la distance moyenne entre le point de données i et toutes les données dans les clusters différents. Supposons maintenant qu'un cluster différent de A soit le cluster C. Ainsi, le calcul de la distance moyenne entre le point de données i et toutes les données dans le cluster C est écrit comme suit :

$$d(i, C) := \frac{1}{|C|} \sum_{j \in C} d(i, j) \tag{2.13}$$

Après avoir calculé $d(i, C)$ pour tous les clusters $C \neq A$, nous sélectionnons ensuite la valeur de distance minimale comme valeur de b_i.

$$b_i = \min_{C \neq A} d(i, C) \tag{2.14}$$

Si le cluster B a la valeur de distance minimale, alors $d(i, B) = b_i$, ce qui est appelé le voisin du point de données i et représente le deuxième meilleur cluster pour le point de

données i après le cluster A. Après avoir déterminé a_i et b_i, le dernier processus consiste à calculer le coefficient de silhouette.

$$s_i = \frac{\max\{a_i, b_i\} - \min\{a_i, b_i\}}{\max\{a_i, b_i\}} \tag{2.15}$$

La valeur de s_i se situe entre -1 et 1, chaque valeur étant interprétée comme suit :

— $s_i \approx 1$: le point de données i est bien classé (dans A).

— $s_i \approx 0$: le point de données i se trouve entre deux clusters (A et B).

— $s_i \approx -1$: le point de données i est mal classé (plus proche du cluster B que de A).

L'Adjusted Rand Score (ARS) est une mesure d'évaluation de similarité entre deux clustering. Il est utilisé pour évaluer la qualité d'un algorithme de clustering en comparant les clusters prédits avec des clusters connus ou véritables. L'ars retourne une valeur entre -1 et 1.

— Une valeur de 1 indique une parfaite similarité entre les deux ensembles de clusters.

— Une valeur proche de 0 ou négative indique un appariement aléatoire ou une dissimilarité entre les ensembles de clusters.

2.6 Conclusion

Dans ce deuxième chapitre, nous avons exploré les méthodes de clustering appliquées aux séries chronologiques. Nous avons discuté des différentes mesures de similarité, y compris la distance euclidienne, qui est simple mais sensible aux décalages et aux distorsions, et la distance dynamique de temps (DTW), qui aligne les séries de manière non linéaire pour capturer les similarités même lorsque les séries présentent des décalages temporels.

Nous avons présenter les principales approches de clustering, telles que le K-means, le clustering hiérarchique et le DBSCAN.

En complément des techniques de base, nous avons exploré les méthodes de validation de clusters, telles que l'indice de silhouette, l'ars. Ces méthodes de validation sont essentielles pour évaluer l'efficacité et la pertinence des résultats du clustering.

Pour conclure, ce chapitre a offert un cadre exhaustif pour comprendre et appliquer les méthodes de clustering aux séries chronologiques. Il a mis en avant la combinaison d'algorithmes robustes, de mesures de similarité appropriées et de techniques de validation rigoureuses.

Chapitre 3

Application des méthodes de Clustering de séries temporelles simulées

3.1 Introduction

Dans ce chapitre, nous abordons l'application des méthodes de clustering aux séries temporelles, en nous concentrons spécifiquement sur l'utilisation du k-means avec quatres distances : distance euclidienne, DTW, ACF, et PACF. Nous avons choisi ces méthodes en raison de leur applicabilité à différents types de données et de leur capacité à révéler des structures sous-jacentes sans supervision préalable. Notre objectif est d'évaluer la qualité des clustering fournis par le k-means avec ces distances, en effectuant des simulations de séries chronologiques, basées sur les modèles : AR, MA, et SARIMA.

Nous commençons par décrire la méthode de sélection et le traitement initial des données simulées, en caractérisant les variétés de séries temporelles élaborées pour imiter des comportements distincts : stables, cycliques et erratiques. Puis, nous détaillons la mise en œuvre de chaque approche de clustering, en insistant sur les méthodes employées pour convertir les séries temporelles en formats compatibles avec les algorithmes de clustering.

Nous discutons en détail des résultats obtenus après avoir appliqué ces méthodes de clustering sur nos données simulées, en examinant la qualité des clusters et en interprétant les groupes formés. La performance de chaque méthode est aussi évaluée avec des critères tels que l'indice de silhouette, l'ars et le temps d'exécution pour mesurer la cohérence au sein des clusters et leur séparation mutuelle.

3.2 Données des simulations

Dans le cadre de notre recherche sur le regroupement des données séquentielles et Afin de tester nos algorithmes, nous avons créé plusieurs séries temporelles synthétiques en utilisant le modèle AR, MA et $SARIMA$. Nous avons varié les paramètres de ces modèles pour simuler différentes dynamiques temporelles. Ensuite, nous avons normalisé les séries temporelles générées afin d'éviter que l'échelle n'influence les résultats du clustering.

Nous avons simulé cent répliques de séries temporelle en utilisant trois modèles : AR, MA et SARIMA. Chaque modèle a été simulé avec trois ensembles de coefficients différents, permettant d'explorer une variété de comportements temporels. Le Tableau 3.1 contient les paramètres des processus simulés.

nous avons chsoisi d'effectuer deux prétraitements aux séries :

1. **rééchantillonage** : ce prétraitement consiste à remplacer la valeur de la série à l'instant t par la moyenne des 8 observations autour de cet instant. Ce prétraitement permet de

Modèle	Paramètres
AR(1)	$\phi = [0.15, -0.35, 0.9]$
AR(2)	$\phi = [0.02, 0.4], [0.9, 0.3], [-0.5, 0.07]$
AR(5)	$\phi = [-0.5, -0.8, -0.2, -0.65, -0.01]$, $\phi = [0.7, 0.25, -0.9, -0.45, 0.1]$, $\phi = [0.35, 0.99, 0.75, 0.21, 0.84]$
MA(1)	$\theta = [0.15, -0.35, 0.9]$
MA(2)	$\theta = [0.02, 0.4], [0.9, 0.3], [-0.5, 0.07]$
MA(5)	$\theta = [-0.5, -0.8, -0.2, -0.65, -0.01]$, $\theta = [0.7, 0.25, -0.9, -0.45, 0.1]$, $\theta = [0.35, 0.99, 0.75, 0.21, 0.84]$
SARIMA (1,1,1)(1,1,1)	$\phi = [0.25, -0.35, 0.9]$, $\theta = [0.7, -0.02, -0.12]$, $\Phi = [0.9, -0.6, 0.88]$, $\Theta = [0.4, -0.3, -0.33]$
SARIMA (2,1,2)(2,1,2)	$\phi = [(0.25, -0.1), (-0.35, 0.15), (0.9, -0.4)]$, $\theta = [(0.7, -0.2), (-0.02, 0.05), (-0.12, 0.03)]$, $\Phi = [(0.9, -0.5), (-0.6, 0.25), (0.88, -0.45)]$, $\Theta = [(0.4, -0.3), (-0.3, 0.2), (-0.33, 0.1)]$
SARIMA (5,1,5)(5,1,5)	$\phi = [(0.25, -0.1, 0.3, -0.2, 0.1), (-0.35, 0.15, -0.2, 0.1, -0.3), (0.9, -0.4, 0.6, -0.3, 0.8)]$, $\theta = [(0.7, -0.2, 0.4, -0.3, 0.2), (-0.02, 0.05, -0.03, 0.08, -0.01), (-0.12, 0.03, -0.1, 0.05, -0.15)]$, $\Phi = [(0.9, -0.5, 0.6, -0.3, 0.4), (-0.6, 0.25, -0.3, 0.2, -0.1), (0.88, -0.45, 0.7, -0.35, 0.9)]$, $\Theta = [(0.4, -0.3, 0.5, -0.2, 0.1), (-0.3, 0.2, -0.25, 0.15, -0.05), (-0.33, 0.1, -0.2, 0.15, -0.25)]$

Tableau 3.1 – Paramètres des différents modèles AR, MA et SARIMA

lisser la série, pour une meilleure visualisation ;

2. **normalisation** : ce prétraitement consiste à centrer et à réduire par l'écart-type de la série. Cela a pour effet de ramener toutes les séries au même ordre de grandeur.

Nous avons choisi d'appliquer les 4 méthodes de clustering suivantes :

- **K-means + distance euclidienne** ;
- **K-means + distance DTW** ;
- **K-means + distance ACF** ;
- **K-means + distance PACF**.

Nous avons choisi d'utiliser l'algorithme K-means en raison de sa simplicité et de sa capacité à identifier des clusters convexes dans l'espace des caractéristiques. Ces méthodes ont été implémentées en utilisant **Python**, notamment avec les bibliothèques **Scikit-Learn** pour K-means avec distance euclidienne et **tslearn** pour la distance DTW. Pour les distances ACF et PACF, nous calculons les représentations par l'ACF et PACF, que nous clusterisons avec le kmeans de **Scikit-Learn**.

3.3 Résultats Numériques

Nous rapportons les résultats des clustering, en nous basant sur l'indice silhouette, ars, et le temps d'exécution. Afin d'alléger la lecture, nous avons choisi de présenter ces résultats avec graphiques. Chaque graphique est constitué des 3 parties suivantes :

1. résultats sur la série brute : la série n'a subit aucun prétraitement ;
2. résultats sur la série rééchantillonée : le clustering est effectué après rééchantillonage des séries ;
3. résultats sur la série normalisée : le clustering est effectué après normalisation des séries.

Nous avons choisi de regrouper les résultats du k-means avec la distance euclidienne et DTW dans la même figure, de même que ceux de l'ACF et PACF. Chaque partie contient 18 résultats sur l'axe des abscisses, correspondant aux 9 types de séries simulées puis clusterisées avec le kmeans avec la distance euclidienne et DTW, ou bien avec ACF et la PACF.
Enfin, nous fournissons les résultats moyens, les pires et les meilleures, dans chacun des cas.

3.3.1 Moyennes des Résultats

L'indice ars montre qu'en moyenne les résultats donnés par l'ACF et PACF sont meilleurs que les distances euclidienne et DTW. La silhouette est approximativement identique, cela montre que cet indice ne permet pas de distinguer le bon clustering. On observe aussi que les résultats des séreis brutes et normalisées est identiques. Le temps d'exécution est plus grand avec les distances euclidienne et DTW. Nous observons également que les séries contenant une composante MA rend difficile le clustering par les distances euclidienne, DTW, et PACF.

Nous pouvons dire qu'en moyenne les distance ACF et PACf sont meilleures, et qu'il n'y a pas lieu d'effectuer les prétraitements.

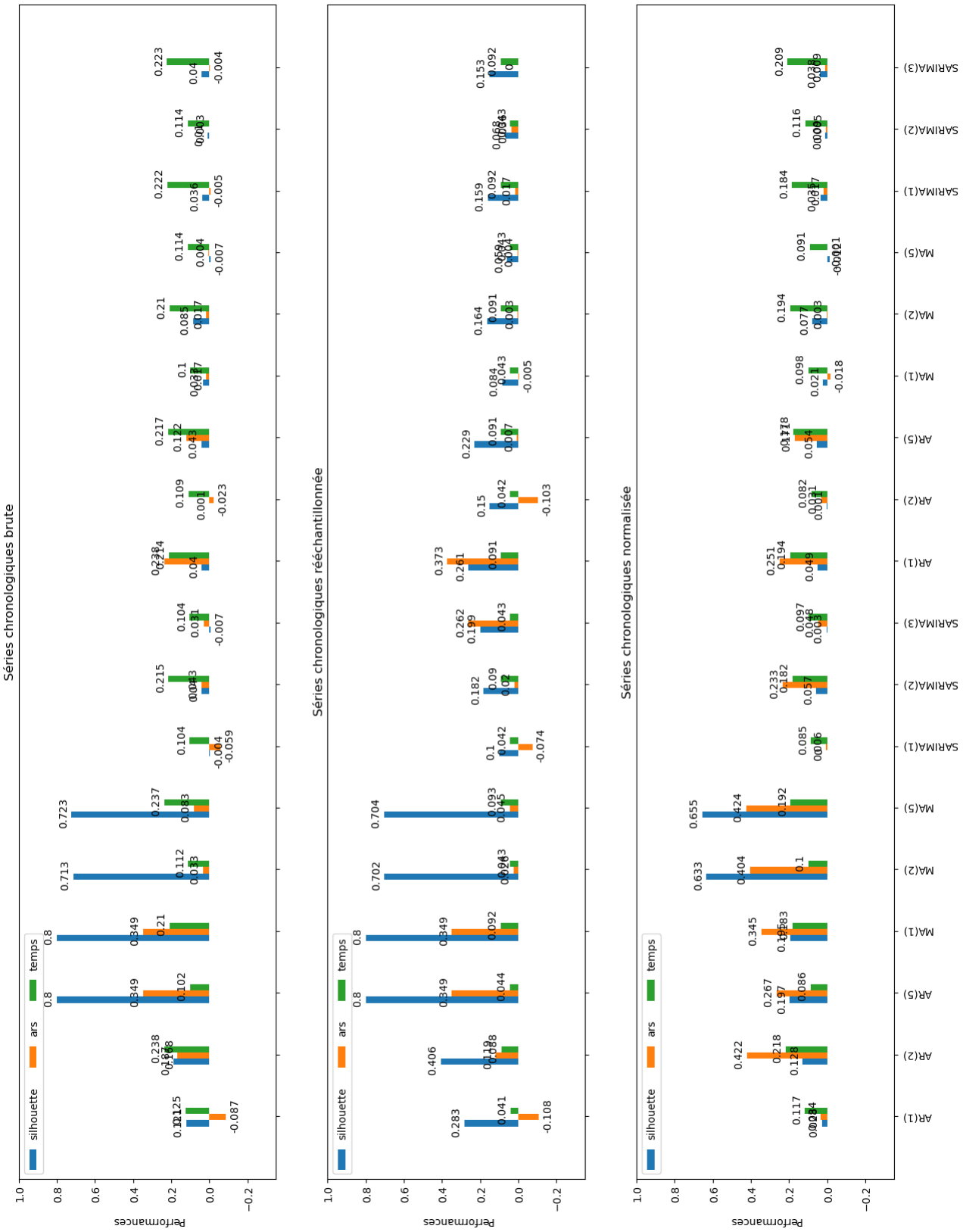

FIGURE 3.1 – Moyennes des résultats obtenus par le K-means. Les 9 premières avec la distance euclidienne et les 9 dernières avec la distance DTW.

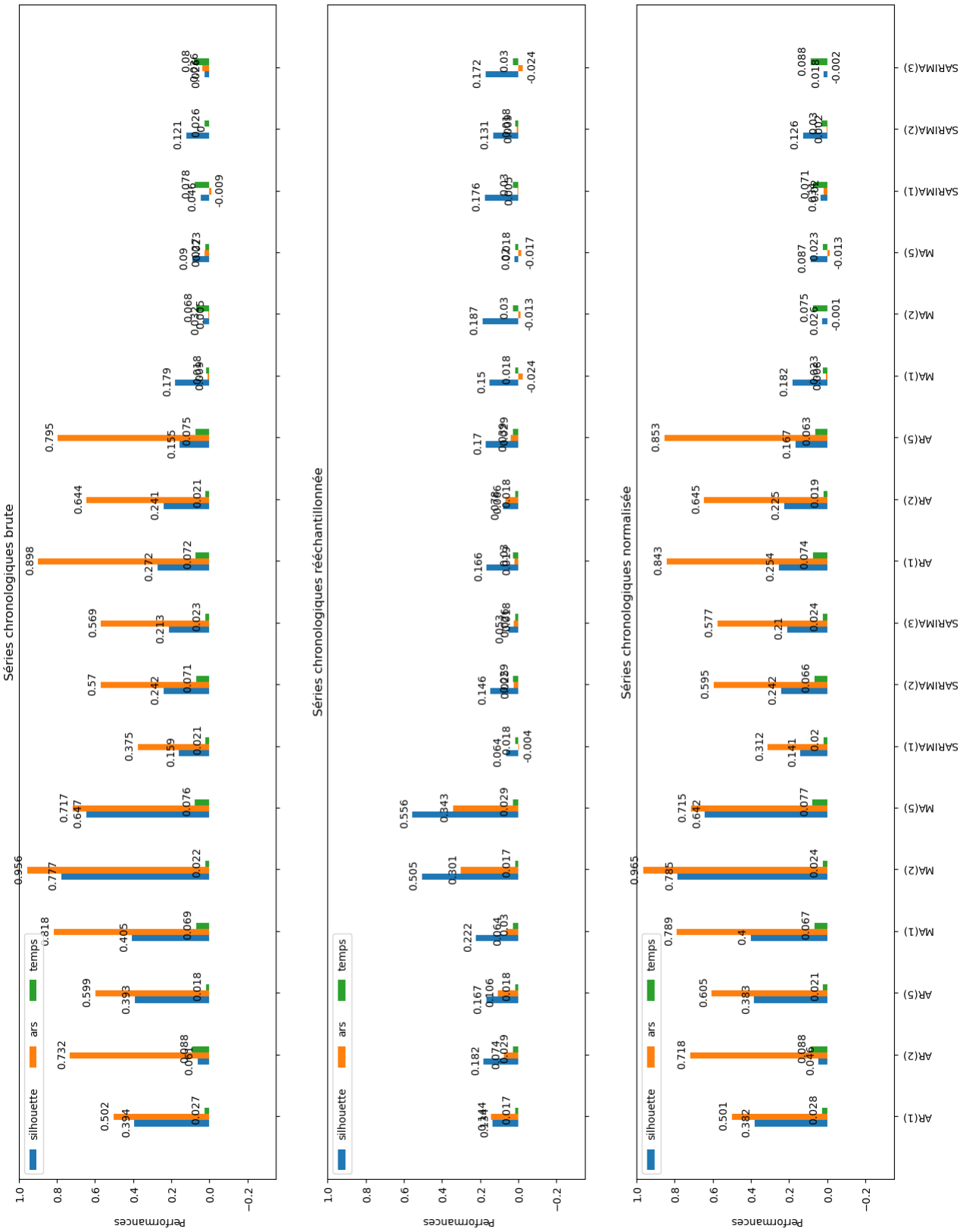

FIGURE 3.2 – Moyennes des résultats obtenus par le K-means. Les 9 premières avec la distance ACF et les 9 dernières avec la distance PACF.

3.3.2 Meilleurs Résultats

Nous observons que dans les meilleurs des cas, le kmeans peut donner un clustering pratiquement parfait. En particulier, quand on utilise les distances ACf et PACF. Néanmoins,

la silhouette n'informe pas suffisamment, car sa grande valeur ne suit pas forcément la tendance de l'ars.

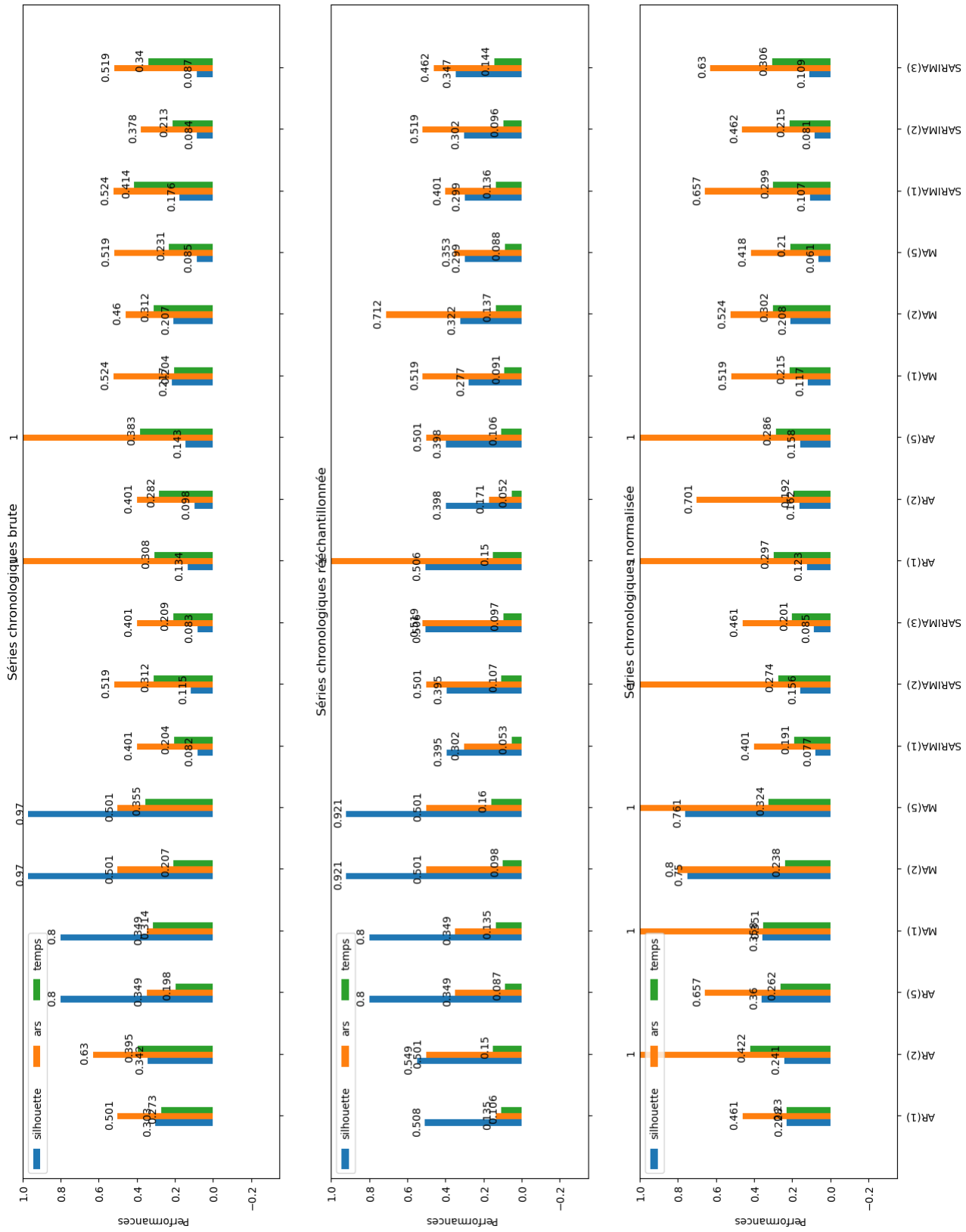

FIGURE 3.3 – Meilleurs résultats obtenus par le K-means. Les 9 premiers avec la distance euclidienne et les 9 derniers avec la distance DTW.

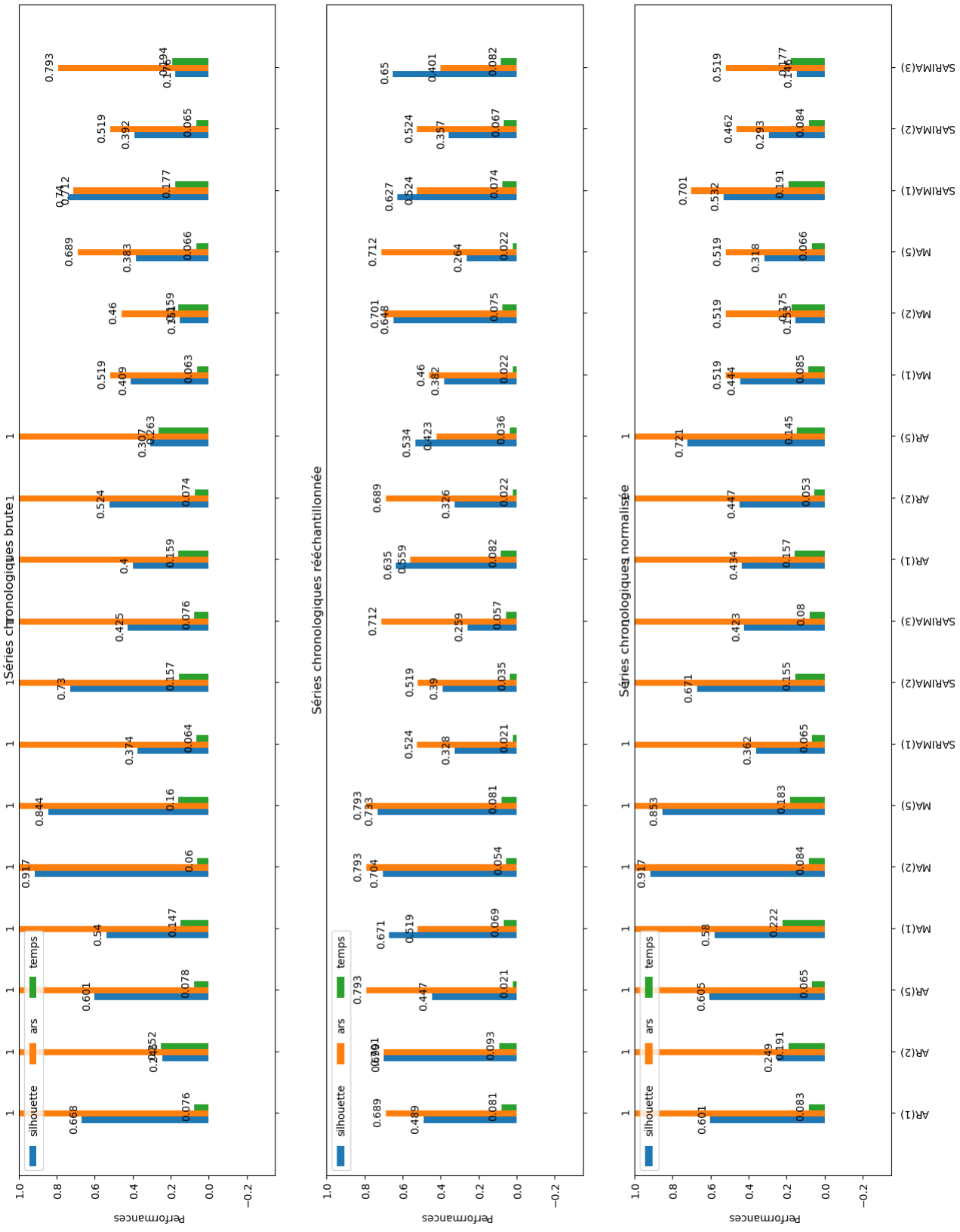

FIGURE 3.4 – Meilleurs résultats obtenus par le K-means. Les 9 premiers avec la distance ACF et les 9 derniers avec la distance PACF.

3.3.3 Pire Résultats

Dans le pire des cas, le kmeans peut échouer à clusteriser des séries chronologiques, et cela ne dépend pas des distances utilisées.

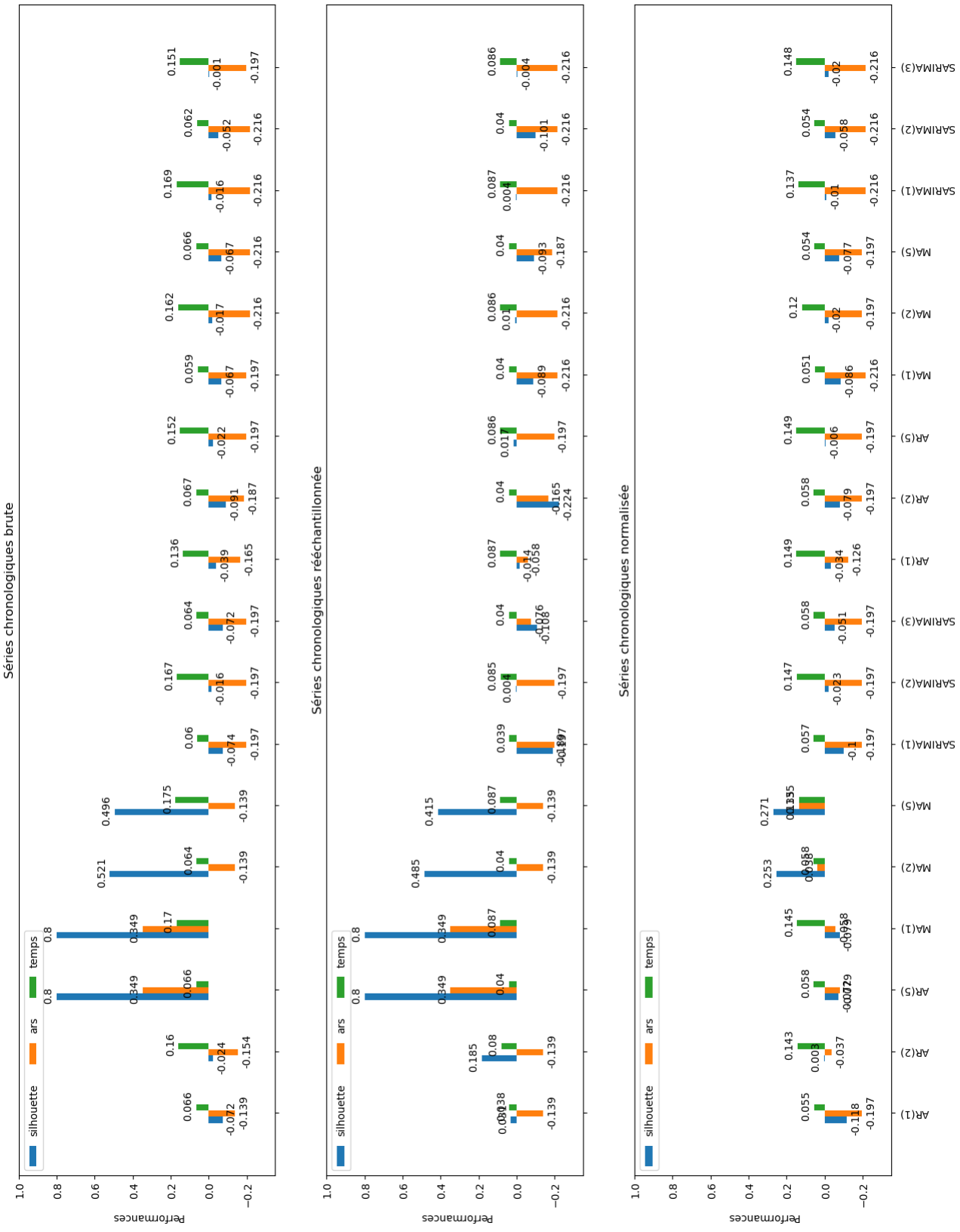

FIGURE 3.5 – Pires résultats obtenus par le K-means. Les 9 premiers avec la distance euclidienne et les 9 derniers avec la distance DTW.

3.4 Comparaison globale des résultats numériques

Cette étude examine et compare les performances de différentes méthodes de regroupement appliquées à des séries chronologiques. Les distances euclidienne, DTW, ACF et PACF sont utilisées pour évaluer la cohérence des clusters, la rapidité des résultats et la qualité des

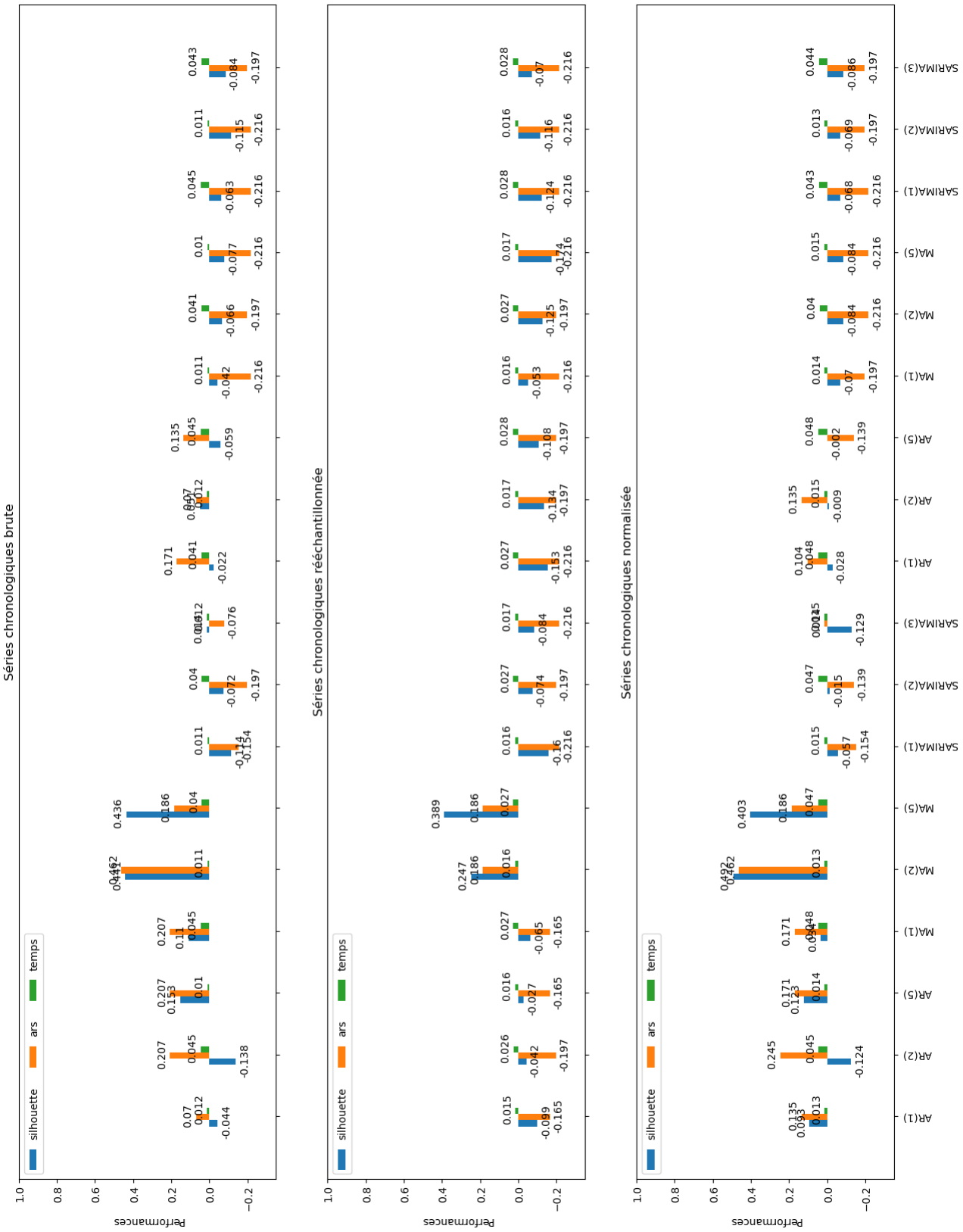

FIGURE 3.6 – Pires résultats obtenus par le K-means. Les 9 premiers avec la distance ACF et les 9 derniers avec la distance PACF.

groupes formés.

1. **Qualité du Clustering :**
   • ACF et PACF offrent une meilleure qualité des clusters par rapport aux distances Euclidienne et DTW.

2. **Rapidité des résultats :**
   • les distance ACF et PACF montrent des temps de calcul plus faibles.
3. **prétraitement de séries temporelles :**
   • La normalisation et le rééchantillonage des séries chronologiques n'apporte rien de particulier à la qualité du clustering.

On remarque aussi que la silhouette des séries $AR(5)$ et $MA(1)$ brutes et rééchantillonées, avec le kmeans + distance euclidienne est restée constante. Cela montre que le clustering obtenu est toujours le même. d'autres simulations n'ont pas confirmé cette observation.

Recommandations : sur la base des simulations effectuées, nous recommandons d'utiliser le kmeans avec les distances ACf et PACF sans prétraitement, de plus l'indice silhouette n'est pas souhaitable pour s'assurer de la qualité du clustering.

3.5 Conclusion

Dans ce chapitre, nous avons exploré l'application de diverses méthodes de clustering sur des séries temporelles simulées. Les distances utilisées pour évaluer la similarité entre les séries incluent les distances euclidienne, DTW, ACF et PACF. Nous avons également pris en compte l'effet du prétraitement des séries temporelles, telles que la normalisation et le rééchantillonage.

Les distances ACF et PACF se sont avérées les plus efficaces pour le clustering, surpassant les distances euclidienne et DTW en termes de qualité et de rapidité. Le prétraitement des séries n'a pas montré d'amélioration significative des résultats. De plus, l'indice silhouette s'est révélé peu pertinent pour évaluer la qualité du clustering.

Les simulations effectuées indiquent que, bien que l'algorithme K-means puisse produire des résultats de clustering presque parfaits dans des conditions idéales, il peut également échouer dans certains cas, indépendamment des distances utilisées.

En conclusion, pour le clustering de séries temporelles simulées, il est recommandé d'utiliser l'algorithme K-means avec les distances ACF et PACF sans prétraitement préalable. Cette approche offre un équilibre optimal entre la qualité du clustering

Conclusion générale

Ce mémoire a exploré les techniques de clustering appliquées aux séries chronologiques, mettant en évidence leur pertinence et leur utilité dans divers domaines d'application. Après avoir examiné les méthodes existantes et mené des expérimentations sur des données simulées, nous avons pu identifier les avantages et les limitations des différentes approches de regroupement.

Dans le premier chapitre, nous avons introduit les concepts théoriques essentiels pour comprendre les séries chronologiques. Nous avons exploré des propriétés essentielles des séries. Ces notions sont cruciales pour analyser les comportements temporels et préparer les données pour des analyses plus complexes. De plus, nous avons présenté les modèles de base, comme les modèles AR, MA et ARMA, ARIMA et SARIMA qui sont fréquemment utilisés pour la modélisation et la prévision des séries chronologiques. Ces modèles fournissent une base théorique pour les méthodes de clustering discutées ultérieurement.

Le deuxième chapitre s'est concentré sur les méthodes de clustering des séries chronologiques. Nous avons examiné en détail les différentes techniques de clustering, notamment le K-means, le clustering hiérarchique, et le DBSCAN, en expliquant leurs principes, leurs avantages, et leurs limitations. Nous avons également mis en lumière l'importance des mesures de similarité, comme la distance euclidienne et la distance dynamique de temps (DTW), pour évaluer la proximité entre les séries. Les techniques de validation, comme l'indice de silhouette, l'ars ont été utilisées pour évaluer la qualité des clusters formés.

Dans le troisième chapitre, nous avons appliqué ces méthodes de clustering à des séries temporelles simulées, créées à partir des modèles AR, MA, et SARIMA. Cette application pratique a permis de tester et de comparer les performances des différents algorithmes. Les résultats ont été évalués à l'aide d'indices de validation tels que l'indice de silhouette, l'ars et le temps d'exécution, mettant en évidence l'importance de la sélection des mesures de similarité et des paramètres de clustering pour obtenir des clusters cohérents et bien séparés.

Ce mémoire apporte des contributions significatives au domaine du clustering des séries chronologiques en fournissant une base théorique, détaillant les concepts et modèles fondamentaux nécessaires pour aborder des analyses temporelles complexes. Il compare les différentes méthodes de clustering, offrant une vue d'ensemble de leurs performances relatives et facilitant le choix des techniques appropriées pour des applications spécifiques. En appliquant ces méthodes à des séries temporelles simulées et en évaluant leur efficacité, il fournit des preuves empiriques précieuses qui permettent d'optimiser ces techniques pour des données réelles.Les résultats obtenus ouvrent la voie à des perspectives futures.

Les travaux futurs pourraient se concentrer sur plusieurs axes d'amélioration et d'exploration. Tout d'abord, il serait bénéfique d'optimiser les algorithmes de clustering en développant et en affinant des méthodes pour renforcer leur robustesse et leur précision, surtout face à des séries temporelles longues et complexes. En parallèle, il serait intéressant d'explorer de nouvelles mesures de similarité adaptées aux spécificités des séries chronologiques, comme les motifs saisonniers et les tendances non linéaires. Par ailleurs, il serait pertinent de combiner le clustering avec d'autres techniques d'analyse des séries chronologiques.

Bibliographie

[Aghabozorgi et. al. 2015] Aghabozorgi, Saeed ; Shirkhorshidi, Ali S. ; Wah, Teh Y. : Time-series clustering–a decade review. In : *Information systems* 53 (2015), S. 16–38

[Ahmed et. al. 2020] Ahmed, Mohiuddin ; Seraj, Raihan ; Islam, Syed Mohammed S. : The k-means algorithm : A comprehensive survey and performance evaluation. In : *Electronics* 9 (2020), Nr. 8, S. 1295

[Arlot 2007] Arlot, Sylvain : *Rééchantillonnage et sélection de modèles*, Université Paris Sud-Paris XI, Diss., 2007

[Bigot 2017] Bigot, Jérémie : Séries chronologiques. In : *Licence* 3 (2017), Nr. 2016, S. 2017

[Brockwell et Davis 1991] Brockwell, Peter J. ; Davis, Richard A. : *Time series : theory and methods*. Springer science & business media, 1991

[Brown 2004] Brown, Robert G. : *Smoothing, forecasting and prediction of discrete time series*. Courier Corporation, 2004

[Caiado et. al. 2006] Caiado, Jorge ; Crato, Nuno ; Peña, Daniel : A periodogram-based metric for time series classification. In : *Computational Statistics & Data Analysis* 50 (2006), Nr. 10, S. 2668–2684

[Daniel et Ruey S. 2021] Daniel, Peña ; Ruey S., Tsay : *Clustering and Classification of Time Series*. Wiley, 2021

[Daudin et. al. 1996] Daudin, JJ ; Duby, C ; Robin, S ; Trécourt, P : Analyse de séries chronologiques. In : *INA-PG, Mathématiques* (1996)

[Ghosal et. al. 2020] Ghosal, Attri ; Nandy, Arunima ; Das, Amit K. ; Goswami, Saptarsi ; Panday, Mrityunjoy : A short review on different clustering techniques and their applications. In : *Emerging Technology in Modelling and Graphics : Proceedings of IEM Graph 2018* (2020), S. 69–83

[Hammouda 2009] Hammouda, Mohamed : *Utilisation des techniques de data mining pour la modélisation du parcours scolaire et la prédiction du succès et du risque d'échec*, Blida, Diss., 2009

[Messaouda 2020] Messaouda, AMMARI : *Détection et Extraction de la tendance et de la saisonnalité dans les séries temporelles*, UNIVERSIT KASDI MERBAH OUARGLA, Diss., 2020

[Moulines et Roueff 2010] Moulines, Eric ; Roueff, François : *Analyse des Séries Temporelles et Applications*. 2010

[Peña et Tsay 2021] *Kapitel* CLUSTERING AND CLASSIFICATION OF TIME SERIES. In : Peña, Daniel ; Tsay, Ruey : *STATISTICAL LEARNING FOR BIG DEPENDENT DATA*. John Wiley & Sons, 2021. – ISBN 9781119417408, S. 211–290

[Prenat et. al. 2010] Prenat, Michel ; Keribin, Christine ; Rossignol, Raphaël : Séries chronologiques. In : *Université Paris-Sud* (2010)

[Rokach et Maimon 2005] Rokach, Lior ; Maimon, Oded : Clustering methods. In : *Data mining and knowledge discovery handbook* (2005), S. 321–352

[SEDDATI 2019] SEDDATI, Lamiae : SÉRIES CHRONOLOGIQUES. (2019)

[Senin 2008] Senin, Pavel : Dynamic time warping algorithm review. In : *Information and Computer Science Department University of Hawaii at Manoa Honolulu, USA* 855 (2008), Nr. 1-23, S. 40

[Touche 2022] Touche, Nassim : *Analyse et Fouille de données financières*. 2022. – Cours de Master 2, Mathématiques Financières

[Zolhavarieh et. al. 2014] Zolhavarieh, Seyedjamal ; Aghabozorgi, Saeed ; Teh, Ying W. : A review of subsequence time series clustering. In : *The Scientific World Journal* 2014 (2014), Nr. 1, S. 312521

Résumé

Ce mémoire se concentre sur le clustering des séries chronologiques, une technique cruciale pour analyser des données temporelles dans divers domaines. Après une introduction aux séries chronologiques et aux modèles de processus comme les AR, MA, ARMA, ARIMA et SARIMA, le mémoire explore différentes méthodes de clustering, y compris l'algorithme K-means et le clustering hiérarchiqueet le DBSCAN.

Les mesures de similarité pour les séries chronologiques, telles que la distance euclidienne et la distance DTW, ainsi que les outils d'évaluation de la qualité du clustering, comme le coefficient silhouette et l'ARS, sont explorés. Des applications pratiques sont présentées à travers la simulation de séries temporelles, avec une analyse comparative des résultats utilisant diverses distances, notamment celles basées sur les fonctions d'autocorrélation (ACF) et d'autocorrélation partielle (PACF).

Le mémoire se termine en mettant en évidence l'importance du clustering des séries chronologiques pour identifier des structures sous-jacentes et détecter des anomalies, fournissant ainsi des outils puissants pour l'analyse et l'interprétation des données temporelles.

Mots-clés :
Série chronologique, clustering, k-means, distance euclidienne, DTW, ACF, PACF.

Abstract

This thesis focuses on time series clustering, a crucial technique for analyzing temporal data across various domains. After introducing time series and process models such as AR, MA, ARMA, ARIMA, and SARIMA, the thesis explores different clustering methods including K-means, hierarchical clustering, and DBSCAN. Similarity measures for time series, such as Euclidean distance and DTW distance, along with clustering quality evaluation tools like silhouette coefficient and ARS, are examined. Practical applications are demonstrated through the simulation of time series, with comparative analysis of results using various distances, particularly those based on autocorrelation functions (ACF) and partial autocorrelation functions (PACF). The thesis concludes by emphasizing the importance of time series clustering in identifying underlying structures and detecting anomalies, thereby providing powerful tools for the analysis and interpretation of temporal data.

Keywords :
Time serie, clustering, k-means, euclidean distance, DTW, ACF, PACF.