Universidade Federal da Paraíba
Centro de Informática
Programa de Pós-Graduação em Modelagem Matemática e Computacional

DISTÂNCIAS ADAPTATIVAS E KERNELIZADAS APLICADAS A AGRUPAMENTO DE SÉRIES TEMPORAIS TIPO INTERVALO

Katy Sylvia Batista Castro

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Modelagem Matemática e Computacional, UFPB, da Universidade Federal da Paraíba, como parte dos requisitos necessários à obtenção do título de Mestre em Modelagem Matemática e Computacional.

Orientadores: Eufrásio de Andrade Lima Neto
Marcelo R. Portela Ferreira

João Pessoa
Março de 2022

Ata da Sessão Pública de Defesa de Dissertação de Mestrado de **KATY SYLVIA BATISTA CASTRO**, candidata ao título de Mestre em Matemática Computacional, na Área de Modelagem Matemática e Computacional, realizada no dia 30 de março de 2022.

Aos trinta dias do mês de março do ano de dois mil e vinte e dois, às 14h, via videoconferência, reuniram-se os membros da Banca Examinadora constituída para julgar o Trabalho Final da discente KATY SYLVIA BATISTA CASTRO**,** vinculada a Universidade Federal da Paraíba sob a matrícula nº 20191026170, candidata ao grau de Mestre em "*Modelagem Matemática e Computacional*", na linha de pesquisa *"Modelagem Probabilística"*, do Programa de Pós-Graduação em Modelagem Matemática e Computacional. A comissão examinadora foi composta pelos professores Eufrásio de Andrade Lima Neto, Orientador e Presidente da Banca; Marcelo Rodrigo Portela Ferreira, Coorientador; Bruno Ferreira Frascaroli, Examinador Interno ao Programa; Telmo de Menezes e Silva Filho, Examinador Externo ao Programa, e Francisco de Assis Tenório de Carvalho, Examinador Externo à Instituição. Dando início aos trabalhos, o Professor Eufrásio de Andrade, Presidente da Banca, cumprimentou os presentes, comunicou aos mesmos a finalidade da reunião e passou a palavra à candidata para que fizesse, oralmente, a exposição do trabalho de dissertação intitulado *"Distâncias adaptativas e kernelizadas aplicadas a agrupamento de séries temporais tipo-intervalo"*. Concluída a exposição, a candidata foi arguida pela Banca Examinadora, que emitiu o seguinte parecer: "***aprovada***". Do ocorrido, eu, Gean Paulo P. M. de Barros, secretário do Programa de Pós-Graduação em Modelagem Matemática e Computacional (PPGMMC), lavrei a presente ata, que vai assinada por mim e pelos membros da Banca Examinadora.

João Pessoa, 30 de março de 2022.

Gean Paulo Pereira Maurício de Barros
Secretário do PPGMMC
SIAPE 2326476

Prof. Dr. Eufrásio de Andrade Lima Neto
Orientador (PPGMMC)

Prof. Dr. Marcelo Rodrigo Portela Ferreira
Coorientador (PPGMMC)

Prof. Dr. Bruno Ferreira Frascaroli
Examinador Interno ao Programa (PPGMMC)

Prof. Dr. Telmo de Menezes e Silva Filho
Examinador Externo ao Programa (UFPB)

Prof. Dr. Francisco de Assis Tenório de Carvalho
Examinador Externo à Instituição (UFPE)

Catalogação na publicação
Seção de Catalogação e Classificação

C355d Castro, Katy Sylvia Batista.
Distâncias adaptativas e Kernelizadas aplicadas a agrupamento de séries temporais tipo / Katy Sylvia Batista Castro. - João Pessoa, 2022.
76 f.

Orientação: Eufrásio de Andrade Lima Neto.
Coorientação: Marcelo Rodrigo Portela Ferreira.
Dissertação (Mestrado) - UFPB/CI.

1. Modelagem matemática. 2. Matemática computacional. 3. Cluster. 4. Séries temporais. 5. Dados tipo intervalo. I. Lima Neto, Eufrásio de Andrade. II. Ferreira, Marcelo Rodrigo Portela. III. Título.

UFPB/BC CDU 519.6(043)

Elaborado por Gracilene Barbosa Figueiredo - CRB-15/794

A minha filha Mel que apesar de ainda não entender o motivo de eu estudar tanto mostra tanta admiração por mim. Sua doçura e carinho nos momentos mais difíceis me ajudaram a ficar firme nessa jornada. Ao deitar ao meu lado enquanto estudo, ela recarrega minhas energias e me dá forças para continuar.

Agradecimentos

Gostaria de agradecer primeiramente a Deus que tornou possível a minha aprovação na seleção. Também a minha doce filha Mel que teve que abrir mão de muitos momentos comigo para que eu pudesse estudar. Ao meu marido Ellan que nunca deixou de ser um companheiro me apoiando e cuidando de nossa filha para que eu pudesse me dedicar mais aos estudos. A minha amiga Mara por ter me dado força para realizar a seleção e por ter acreditado no meu potencial quando nem eu mesma acreditava. A minha nova amiga Emília, uma irmã que esse mestrado me deu e que mesmo sem me conhecer me ajudou em vários momentos. Aos meus orientadores professores Eufrásio e Marcelo que sempre foram muito compreensivos com meus horários e dificuldades e sempre se dispuseram a me ajudar, mesmo tarde da noite, aos finais de semana.

Resumo da Dissertação apresentada ao PPGMMC/CI/UFPB como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

DISTÂNCIAS ADAPTATIVAS E KERNELIZADAS APLICADAS A AGRUPAMENTO DE SÉRIES TEMPORAIS TIPO INTERVALO

Katy Sylvia Batista Castro

Março/2022

Orientadores: Eufrásio de Andrade Lima Neto
Marcelo R. Portela Ferreira

Programa: Modelagem Matemática e Computacional

A tarefa de agrupar faz parte do cotidiano e da natureza humana. A literatura que trata de agrupamentos disponibiliza técnicas, métricas e algoritmos para realizar essa tarefa. Em particular, o agrupamento de dados observados ao longo do tempo e em forma de intervalos representa um desafio, com novos métodos sendo propostos para essa finalidade. A vantagem das distâncias adaptativas é que elas atribuem pesos diferentes às variáveis do agrupamentos, e um algoritmo que consegue se adaptar a isso pode trazer resultados muito superiores aos algoritmos que tratam todas as variáveis da mesma forma, com o mesmo nível de importância. Ademais, a kernelização torna possível trabalhar com dados em um novo espaço, diferente do espaço original, onde os grupos venham a apresentar uma melhor separação. O objetivo deste trabalho é considerar novas distâncias para o método K-Means no agrupamento de séries temporais de dados tipo intervalo. Utilizaremos distâncias adaptativas e distâncias calculadas através da kernelização da métrica e do espaço de características. Para validar os algoritmos propostos realizamos um estudo com séries temporais geradas a partir dos parâmetros de modelos Autorregressivo Espaço-Tempo (STAR, do inglês *Space-Time Autoregressive*), utilizando simulações Monte Carlo, bem como dados reais. A comparação dar-se-á através de índices externos e internos. Os resultados obtidos nas simulações demonstram que os algoritmos propostos apresentaram desempenho superior em relação aos métodos existentes. A aplicação a dados reais considerou séries de criptomoedas e índices tradicionais como ouro, petróleo, bolsas de valores, entre outros. Os resultados apontam *insights* que poderão ser usados para trabalhos futuros na área de aprendizagem de máquina e economia.

Abstract of Dissertation presented to PPGMMC/CI/UFPB as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

KERNELIZED AND ADAPTIVE DISTANCES FOR CLUSTERING INTERVAL TIME SERIES

Katy Sylvia Batista Castro

March/2022

Advisors: Eufrásio de Andrade Lima Neto
Marcelo R. Portela Ferreira

Program: Computational Mathematical Modelling

The task of clustering is part of everyday life and human nature. The literature that deals with clustering provides techniques, metrics and algorithms to accomplish this task. In particular, the clustering of observed data over time and in the form of intervals represents a challenge, with new methods being proposed for this purpose. The advantage of adaptive distances is that they assign different weights to the variables of clusters, and an algorithm that succeeds in adapting to this can bring results far superior to algorithms that treat all variables in the same way, with the same level of importance. Moreover, kernelization makes it possible to work with data in a new space, different from the original space, where the groups will present a better separation. The objective of this work is to consider new distances for the K-Means method in the clustering of interval time series. We will use adaptive distances and distances calculated through the kernelization of the metric and the feature space. To validate the proposed algorithms, we performed a study with time series generated from the parameters of Space-Time Autoregressive (STAR) models, using Monte Carlo simulations as well as real data. The comparison will take place through external and internal indices. The results obtained in the simulations demonstrate that the proposed algorithms performed better than the existing methods. The application to real data considered cryptocurrency series and traditional indices such as gold, oil, stock exchanges, among others. The results point to *insights* that can be used for future work in machine learning and economics.

Sumário

Lista de Figuras **x**

Lista de Tabelas **xi**

1 Introdução **1**

1.1 Motivação 1

1.2 Objetivos: Geral e Específicos 3

2 Estado da Arte **4**

2.1 Séries temporais 4

2.1.1 Função de autocorrelação 4

2.1.2 Modelos autorregressivos e médias móveis 5

2.2 Métodos de agrupamento 7

2.2.1 Agrupamento, métodos de distância e ordenação 7

2.2.2 Métodos não hierárquicos de agrupamento 8

2.3 Agrupamento de séries temporais 11

2.3.1 Formulação do problema 11

2.3.2 Medidas de dissimilaridade no agrupamento de séries temporais 12

2.3.3 Algoritmos de agrupamento de séries temporais 13

2.3.4 Medidas de avaliação de agrupamento de séries temporais . . . 14

2.4 Agrupamento de séries temporais tipo-intervalo 15

2.4.1 Variáveis tipo-intervalo 16

2.4.2 Séries temporais tipo-intervalo 16

2.4.3 Distâncias para dados tipo-intervalo 17

2.4.4 Agrupamento de séries temporais tipo-intervalo utilizando o método K-Means 17

2.5 Funções Kernel 21

2.5.1 Funções de *kernel* para intervalos 22

2.6 Métodos de agrupamento para intervalos 23

2.6.1 Kernel K-Means baseado na Kernelização da métrica 23

2.6.2 Kernel K-Means no espaço de características 25

2.7 Agrupamento baseado em Distâncias Adaptativas 26

3 Métodos Propostos **32**

3.1 Novos Algoritmos de Agrupamento de Séries Temporais Intervalares . 33

3.1.1 Agrupamento baseado em descritores estatísticos 33

3.1.2 Agrupamento baseado nos coeficientes do modelo STAR . . . 35

4 Resultados e Discussões **38**

4.1 Estudos de simulação 38

4.1.1 Resultados dos experimentos 42

4.2 Aplicação a dados reais 46

4.2.1 Resultados da aplicação 47

5 Conclusão **60**

5.1 Trabalhos Futuros 61

Referências Bibliográficas **62**

Lista de Figuras

4.1 Apresentação de duas séries temporais tipo-intervalo representando a configuração do Cenário 1 39
4.2 Apresentação de duas séries temporais tipo-intervalo representando a configuração do Cenário 2 40
4.3 Apresentação de duas séries temporais tipo-intervalo representando a configuração do Cenário 3 40
4.4 Apresentação de duas séries temporais tipo-intervalo representando a configuração do Cenário 4 41
4.5 Representação gráfica do agrupamento dos índices financeiros obtidos pelo método Kernel K-means - Dados mensais de 15 anos 49
4.6 Representação gráfica do agrupamento dos índices e criptomoedas obtidos pelo método Kernel K-means - Dados diários de um ano . . . 52
4.7 Representação gráfica do agrupamento dos índices e criptomoedas obtidos pelo método Kernel K-means - Dados semanais de um ano . 55
4.8 Representação gráfica do agrupamento dos índices e criptomoedas obtidos pelo método Kernel K-means - Dados mensais de um ano . . 58

Lista de Tabelas

4.1 Cenário 1: média e desvio-padrão (DP) para o índice de Rand ajustado segundo a abordagem, configurações dos clusters e o método de agrupamento. 42
4.2 Cenário 2: média e desvio-padrão (DP) para o índice de Rand ajustado segundo a abordagem, configurações dos clusters e o método de agrupamento. 43
4.3 Cenário 3: média e desvio-padrão (DP) para o índice de Rand ajustado segundo a abordagem, configurações dos clusters e o método de agrupamento. 44
4.4 Cenário 4: média e desvio-padrão (DP) para o índice de Rand ajustado segundo a abordagem, configurações dos clusters e o método de agrupamento. 45
4.5 Dados mensais de 15 anos: segundo a abordagem, configurações dos clusters e o método de agrupamento. 48
4.6 Ranking dos métodos nos índices de comparação das séries de índices financeiros de 15 anos. 49
4.7 Agrupamentos dos índices financeiros obtidos pelo método Kernel K-means - Dados mensais de 15 anos. 50
4.8 Dados diários de um ano: segundo a abordagem, configurações dos clusters e o método de agrupamento. 51
4.9 Ranking dos métodos nos índices de comparação das séries diárias. . . 51
4.10 Agrupamentos dos índices financeiros e criptomoedas obtidos pelo método Kernel K-means - Dados diários de um ano. 53
4.11 Dados semanais de um ano: segundo a abordagem, configurações dos clusters e o método de agrupamento. 54
4.12 Ranking dos métodos nos índices de comparação das séries semanais. 55
4.13 Agrupamentos dos índices financeiros e criptomoedas obtidos pelo método K-means - Dados semanais de um ano. 56
4.14 Dados mensais de um ano: segundo a abordagem, configurações dos clusters e o método de agrupamento. 57

4.15 Ranking dos métodos nos índices de comparação das séries mensais. . 58
4.16 Agrupamentos dos índices financeiros e criptomoedas obtidos pelo método K-means - Dados mensais de um ano. 59

Capítulo 1

Introdução

1.1 Motivação

O desenvolvimento tecnológico permitiu uma maior facilidade de coleta de informações, fazendo com que surgisse um aumento no volume de dados disponíveis para análise. Estes sendo gerados nas mais variadas áreas não só da ciência, mas da vida cotidiana, o que trouxe a necessidade de melhorar e desenvolver técnicas de tratamento e análise de dados. Um exemplo relativamente novo é o conceito das criptomoedas que geram todos os dias informações que podem ser, inclusive, extraídas em formato de séries temporais (BROCKWELL e DAVIS, 2002). No campo da análise de dados, as técnicas de agrupamento vêm sendo estudadas há muitos anos, afinal, agrupar é uma solução para a classificação de conjuntos de dados grandes quando não sabemos nada previamente sobre estes conjuntos. Além disso, agrupar traz vantagens e celeridade na tomada de decisão, já que é muito mais rápido tomar uma decisão e uma ação em relação a um grupo do que a cada item separadamente. Entretanto, é importante que os dados estejam agrupados utilizando um método de agrupamento eficiente (XU e WUNSCH, 2005). Por exemplo, o agrupamento de séries temporais pode trazer a possibilidade de realizar predições sobre grupos de variáveis de forma muito mais rápida e eficiente (AGHABOZORGI et al., 2015).

Os dados tipo-intervalo (BILLARD e DIDAY, 2006) são utilizados quando a informação pontual, por exemplo, da média, do valor mínimo ou do valor máximo de uma variável aleatória não são suficientes para representar o fenômeno em estudo. Assim, um dado tipo-intervalo pode enriquecer a análise, como seria o caso, por exemplo, da medição diária de temperaturas ou da variação diária de uma ação da bolsa de valores.

A motivação deste trabalho surge em função da complexidade de alguns conjuntos de dados como, por exemplo, padrões não-linearmente separáveis, nos quais os métodos de agrupamento que utilizam as distâncias tradicionais têm eficácia li-

mitada (FERREIRA, 2013). Além disso, métodos de agrupamentos que utilizam as distâncias tradicionais, muitas vezes, também não conseguem dar a importância devida a cada variável, já que cada uma pode trazer uma contribuição diferente para o agrupamento. Dessa forma, algoritmos de agrupamento não-supervisionados que utilizam distâncias adaptativas podem resolver tal problema (FERREIRA et al., 2020).

Tornou-se então imprescindível a busca por métodos de agrupamento que suportassem conjuntos de dados grandes, porém com um baixo custo operacional. As séries temporais trazem dados ao longo do tempo, como por exemplo vendas, medições da temperatura em determinada cidade, número de passageiros que utilizam transporte público, produção de alimentos, entre outros. Em diversas situações, é preciso agrupar um grande número de séries temporais. Essa análise torna mais prática a tomada de decisão, já que decidir sobre um grupo é mais prático do que decidir sobre os itens separadamente.

O objetivo principal deste trabalho será apresentar alternativas de distâncias para agrupamento de séries temporais para dados tipo-intervalo, focando principalmente nas distâncias adaptativas e kernelizadas aplicadas aos métodos propostos por Maharaj, Teles, e Brito (2019).

Visando obter um melhor agrupamento, a utilização de novas distâncias pode trazer resultados interessantes, principalmente, quando o conjunto de dados é muito grande. A avaliação de desempenho entre os métodos irá considerar os índices como Rand Ajustado para os dados simulados e Silhueta, Conectividade e Dunn (ARBELAITZ et al., 2013) para os dados reais.

A organização desta dissertação se dará da seguinte forma: no Capítulo 2, trazemos uma revisão bibliográfica sobre os principais temas que serão abordados neste trabalho, a saber: séries temporais na seção 2.1, métodos de agrupamento na seção 2.2, agrupamento de séries temporais na seção 2.3 e, finalmente, na seção 2.4 o agrupamento de séries temporais tipo-intervalo. O Capítulo 3 traz a apresentação do modelo proposto e algumas distâncias, a saber: (i) distâncias calculadas a partir da kernelização da métrica, (ii) distâncias calculadas a partir da kernelização do espaço de características e (iii) distâncias adaptativas. No Capítulo 4, iniciamos com a aplicação dos métodos propostos em dados simulados. Apresentamos os resultados através de gráficos, tabelas e análises e trazemos os resultados obtidos. Finalmente, no Capítulo 5 apresentamos as conclusão do trabalho e sugestões de trabalhos futuros.

1.2 Objetivos: Geral e Específicos

O objetivo geral deste trabalho é apresentar novos métodos de agrupamento para séries temporais tipo-intervalo baseados na kernelização da métrica, do espaço de características e em distâncias adaptativas.

Ademais, de modo a alcançar o objetivo geral, apresento os seguintes objetivos específicos:

- Proposição, desenvolvimento, implementação e aplicação de um algoritmo para agrupamento não-hierárquico baseado na kernelização da métrica de descritores estatísticos de séries temporais tipo-intervalo;
- Proposição, desenvolvimento, implementação e aplicação de um algoritmo para agrupamento não-hierárquico baseado na kernelização do espaço de características dos descritores de Séries Temporais tipo intervalo;
- Proposição, desenvolvimento, implementação e aplicação de um algoritmo para agrupamento não-hierárquico baseado nas medidas adaptativas para os descritores de Séries Temporais tipo intervalo;
- Proposição, desenvolvimento, implementação e aplicação de um algoritmo para agrupamento não-hierárquico baseado na kernelização da métrica das estimativas dos parâmetros em modelos de espaço-tempo;
- Proposição, desenvolvimento, implementação e aplicação de um algoritmo para agrupamento não-hierárquico baseado na kernelização do espaço de características das estimativas dos parâmetros em modelos de espaço-tempo;
- Proposição, desenvolvimento, implementação e aplicação de um algoritmo para agrupamento não-hierárquico baseado nas medidas adaptativas para das estimativas dos parâmetros em modelos de espaço-tempo;
- Avaliação do desempenho dos métodos em dados sintéticos e dados reais, utilizando índices comparativos internos e externos.

Capítulo 2

Estado da Arte

Neste capítulo, apresentaremos uma revisão dos temas abordados neste trabalho necessários para o desenvolvimento dos métodos de agrupamento propostos para séries temporais tipo-intervalo, a saber: séries temporais, métodos de agrupamento para dados usuais, dados tipo-intervalo e agrupamento de séries temporais.

2.1 Séries temporais

Cowpertwait e Metcalfe (2009) definem séries temporais como dados dispostos em ordem cronológica. A partir do estudo de séries temporais é possível predizer o comportamento futuro de um fenômeno de interesse, baseado nas informações passadas contidas na série. Uma série temporal não é uma simples sequência temporal, mas a organização de dados dinâmicos que mudam em função do tempo. Podemos utilizar os modelos de séries temporais para entender o comportamento de fenômenos de interesse e realizar predições.

Algumas definições são recorrentes quando se trata de séries temporais, uma delas é o Ruído Branco (*white noise*). Quando não se captura nenhuma estrutura de dependência entre as observações em uma série temporal, temos um ruído branco. Espera-se que os resíduos de um modelo de séries temporais apresentem tal comportamento.

2.1.1 Função de autocorrelação

A partir da correlação presente nas séries temporais, podemos verificar o grau de dependência existente entre as observações. A função de autocorrelação (acf, do inglês *autocorrelation function*), é definida por:

$$\gamma_k = E[(x_t - \mu)(x_{t+k} - \mu)], \tag{2.1}$$

onde x_t é a observação no tempo t, x_{t+k} é a observação no tempo t+k e μ é a média de x. Assim, o gráfico da função de autocorrelação pode trazer informações importantes acerca dos lags k que apresentam uma correlação significativa para a série. No tempo zero, a acf será 1 visto que o parâmetro $k = 0$ representa a comparação da observação x_t com ela própria.

É possível também analisarmos a série temporal pela sua correlação parcial (pacf, do inglês *partial autocorrelation function*), onde os efeitos das correlações das menores defasagens são removidos.

A linguagem R oferece vários pacotes e comandos para tratar dados de séries temporais. A plotagem do correlograma, inclusive, ajuda a identificar os pontos muito próximos de zero.

2.1.2 Modelos autorregressivos e médias móveis

Uma classe importante de modelos que utilizamos para analisar séries temporais são chamados de modelos autorregressivos (AR).

Um modelo autorregressivo de ordem p - AR(p) - é definido por:

$$x_t = \alpha_1 x_{t-1} + \alpha_2 x_{t-2} + \ldots + \alpha_p x_{t-p} + w_t, \tag{2.2}$$

onde w_t é um ruído branco.

Em Cowpertwait e Metcalfe (2009), encontramos que um processo só é estacionário se os valores absolutos de todas as raízes da equação característica excederem uma unidade.

As séries estacionárias estritas são assim chamadas quando uma mudança de tempo arbitrária não modifica sua distribuição conjunta. Isso implica que, a média e a variância são constantes no tempo e que, a covariância entre as séries (original e com a modificação) vai depender apenas da defasagem de tempo de ambas. Note, que sempre que uma série é estritamente estacionária, a situação acima ocorre, mas o contrário não é necessariamente verdade. Ou seja, é possível uma série temporal ter média e a variância constantes no tempo, a covariância depender apenas da defasagem e a série não ser estritamente estacionária. Nesse caso ela será chamada de estacionária de segunda ordem.

O modelo de médias móveis, também conhecido como processo MA(do inglês *Moving Average*), consiste em uma soma finita dos termos de um ruído branco (white noise) estacionário. Em geral, um processo MA é invertível se todas as raízes da equação característica tiverem seu valor absoluto maior que um. O modelo estacionário Médias Móveis de ordem q - MA(q) - é definido por:

$$x_t = w_t + \beta_1 w_{t-1} + \beta_2 w_{t-2} + \ldots + \beta_q w_{t-q}, \tag{2.3}$$

onde o w_t é o ruído branco com média zero e variância σ^2.

Métricas como o AIC (do inglês *Akaike Information Criterion*, (Akaike, 1974)) e o BIC (do inglês *Bayesian Information Criterion*, (Schwarz, 1978)) ajudam a escolher o modelo mais indicado, já que eles penalizam modelos com muitos parâmetros que não acrescentem performance. A diferença é que o BIC penaliza mais fortemente modelos mais complexos.

Modelos mistos: processo ARMA

A sigla ARMA vem da tradução em inglês da expressão *Autoregressive–Moving-Average*. De forma simples, se termos autorregressivos e médias móveis são encontrados juntos, então podemos considerar que o processo é ARMA(p,q), onde p e q são as respectivas ordens autorregressivas e média móveis. Definimos um processo Autorregressivo Médias Móveis ARMA(p,q) por:

$$x_t = \alpha_1 x_{t-1} + \alpha_2 x_{t-2} + \ldots + \alpha_p x_{t-p} + w_t + \beta_1 w_{t-1} + \beta_2 w_{t-2} + \ldots + \beta_q w_{t-q}. \quad (2.4)$$

Segundo Cowpertwait e Metcalfe (2009) é importante notar os seguintes pontos:

- O processo ARMA será estacionário se todas as raízes da parte autorregressiva não excederem uma unidade.
- O processo ARMA será invertível se todas as raízes da parte das médias móveis não excederem uma unidade.
- O processo AR(p) é um caso especial de ARMA assim como o MA(q).

Modelo autorregressivo integrado de médias móveis - processo ARIMA

Diferente do ARMA, o ARIMA é um modelo não estacionário, sendo de grande utilidade quando não é possível transformar a série em estacionária. Definimos o processo ARIMA(p,q,d) como:

$$\theta_p(B)(1-B)^d x_t = \phi_q(B) w_t, \quad (2.5)$$

onde θ_p e ϕ_q são polinômios de ordem p e q respectivamente e B é o operador de retrocesso. Esse processo também tem variações e pode ser estendido para o ARIMA sazonal ou para os modelos ARCH (heteroscedasticidade condicional autoregressiva).

2.2 Métodos de agrupamento

Muitas vezes trabalhamos com uma grande quantidade de informações que possuem semelhanças entre si. Para que possamos analisar de forma mais rápida e até mesmo tomar ações sobre tais informações, realizamos o agrupamento delas. Como o assunto principal deste trabalho será sobre agrupamento de séries temporais, temos, antes de tudo, que entender o que é e como são feitos os agrupamentos de dados simples.

Antes, porém, faz-se necessário realizar uma breve explanação sobre Distâncias, já que vamos agrupar as observações através delas e podemos usar vários tipos de distâncias e formas de calculá-las.

2.2.1 Agrupamento, métodos de distância e ordenação

Agrupar significa organizar observações (indivíduos, imagens, formas, pixels, etc.) de modo que observações pertencentes a um mesmo grupo têm um alto grau de similaridade enquanto que observações em grupos diferentes têm um alto grau de dissimilaridade. A medida de dissimilaridade (ou de similaridade) desempenha um papel fundamental em métodos de agrupamento. Medidas de distâncias são exemplos importantes de medidas de dissimilaridade. Nesta seção, apresentaremos alguns conceitos básicos sobre medidas de distância.

Distâncias

Sejam dois vetores $\boldsymbol{x}_i$ e $\boldsymbol{y}_i \in \mathbb{R}^p$ e a matriz $\boldsymbol{A}$ positiva, a distância entre $\boldsymbol{x}_i$ e $\boldsymbol{y}_i$ será então dada pela expressão:

$$d^2(\boldsymbol{x}_i, \boldsymbol{y}_i) = |\boldsymbol{x}_i - \boldsymbol{y}_i|^2_{\boldsymbol{A}} = (\boldsymbol{x}_i - \boldsymbol{y}_i)'\boldsymbol{A}(\boldsymbol{x}_i - \boldsymbol{y}_i), \tag{2.6}$$

em que a a matriz $\boldsymbol{A}$ é chamada métrica.

Sejam $\boldsymbol{x}_i, \boldsymbol{y}_i$ e $\boldsymbol{z}_i \in \mathbb{R}^p$, então as seguintes propriedades são verificadas para a função d(·,·):

- $d(\boldsymbol{x}_i, \boldsymbol{y}_i) \geq 0, \forall \boldsymbol{x}_i \neq \boldsymbol{y}_i$;
- $d(\boldsymbol{x}_i, \boldsymbol{y}_i) = 0, \forall \boldsymbol{x}_i = \boldsymbol{y}_i$;
- $d(\boldsymbol{x}_i, \boldsymbol{y}_i) \leq d(\boldsymbol{x}_i, \boldsymbol{z}_i) + d(\boldsymbol{y}_i, \boldsymbol{z}_i)$.

O uso de medidas de dissimilaridade tem o objetivo de reduzir ao máximo a subjetividade do método de agrupamento. A distância Euclidiana é a medida de dissimilaridade mais comumente utilizada e fornece a distância em linha reta entre

dois pontos. Se considerarmos $\boldsymbol{A} = \boldsymbol{I}$, então a equação (2.6) resulta na distância Euclidiana ao quadrado:

$$d^2(\boldsymbol{x}_i, \boldsymbol{y}_i) = (\boldsymbol{x}_i - \boldsymbol{y}_i)'(\boldsymbol{x}_i - \boldsymbol{y}_i) = \sum_{j=1}^{p}(x_{ij} - y_{ij})^2. \tag{2.7}$$

Ainda, se sendo $\boldsymbol{A} = \boldsymbol{S}^{-1}$, em que $\boldsymbol{S}$ é a matriz de variância/covariância amostral, então, temos a distância estatística ou distância de Mahalanobis, dada por:

$$d^2(\boldsymbol{x}_i, \boldsymbol{y}_i) = (\boldsymbol{x}_i - \boldsymbol{y}_i)'\boldsymbol{S}^{-1}(\boldsymbol{x}_i - \boldsymbol{y}_i). \tag{2.8}$$

Analisando minuciosamente, vemos que essa distância se encontra no expoente da equação da função densidade de probabilidade da distribuição Normal Multivariada, representando assim a distância entre a observação multivariada $\boldsymbol{x}$ e a média $\boldsymbol{\mu}$ da variável aleatória multivariada. Isso mostra a relação entre estes dois conceitos tão importantes.

2.2.2 Métodos não hierárquicos de agrupamento

Os métodos não hierárquicos, ou de partição, como também são conhecidos, diferem dos métodos hierárquicos pela necessidade (em sua maioria) da definição do número de grupos K *a priori*. Há algumas exceções como, por exemplo, o DBScan e o Mean Shift que apesar de particionais não precisarem da definição prévia do número de grupos. Em geral, métodos não hierárquicos partem de uma partição inicial aleatória e alternam entre uma etapa de representação e outra de alocação até a convergência, quando nenhuma observação é realocada entre os grupos. As etapas de definição dos representantes (centróides, medóides, etc.) dos grupos e definição da melhor partição são estabelecidas em um processo de otimização, através da minimização de uma função objetivo adequada. O método DBScan é uma exceção, já que não possui representantes e suas etapas são diferentes.

Em resumo, os métodos de partição mais conhecidos são:

- K-Means $\rightarrow$ É o método não hierárquico mais utilizado. Ele foi proposto por Macqueen (1967). Divide os itens em um número pré-determinado de grupos, e na hora de calcular as distâncias entre os itens e os grupos, o centroide definido será a média. A desvantagem é que o número de grupos nem sempre será conhecido antes de iniciar o processo. Esse será o método que nos aprofundaremos mais.

- Método Fuzzy K-Means $\rightarrow$ Neste método, cada observação pode pertencer a todos os grupos com um certo grau de pertinência, chamado de grau de

pertinência *fuzzy*. Em alguns casos é um método muito útil, já que nem sempre é possível alocar um objeto em apenas um grupo. Um bom exemplo disso está na segmentação de clientes. Uma empresa que trabalha com departamentos variados pode ter um cliente que goste de esportes, mas também goste de jogos digitais. Agrupá-lo apenas em esportes para o envio de publicidade pode fazer a empresa perder oportunidade de vendas, mas definindo o coeficiente de relação desse cliente com cada departamento pode definir a frequência de publicidade de cada departamento que será enviado a ele. (DUNN, 1973)

- Método PAM (do inglês *Partition Around Medoids)* → A maioria dos métodos de partição basicamente se iniciam da mesma forma. Assim como o método K-Means, os itens são agrupados aleatoriamente e o representante é definido. No caso do método PAM, são encontrados os medoides de cada um dos grupos. Depois de encontrar os medoides, cada objeto será relacionado com os seus grupos e com os demais grupos e então é alocado no grupo mais próximo. Seu custo computacional é alto para muitos objetos. Porém, sofre menos influência dos outliers. Compara a dissimilaridade entre os objetos dentro do grupo com os objetos fora do grupo, quanto mais próximo de 1, melhor classificado o objeto está. (KAUFMAN e ROUSSEAU, 2005)

- Método CLARA(do inglês *Clustering Large Applications*) → Este método utiliza amostras dos dados no lugar de utilizar o conjunto inteiro. Então, após realizar o agrupamento da amostra pelo método PAM, ele vai alocando os outros itens nos cluster já existentes. Isso é computacionalmente mais barato porque utiliza matriz de medidas e não matriz de similaridades. (KAUFMAN e ROUSSEAU, 2005)

Este trabalho irá considerar apenas os métodos de partição. Dessa forma, métodos hierárquicos de agrupamento como o Ward's (WARD, 1963), o AGNES(do inglês *Agglomerative Nesting*) (KAUFMAN e ROUSSEAU, 2005) e o DIANA(do inglês *Divisive Analysis*) (KAUFMAN e ROUSSEAU, 2005) não serão considersdos nesta dissertação.

Método de partição K-Means

Além de ser o método não-hierárquico mais utilizado, o K-Means serve como base para os métodos desenvolvidos nesse trabalho.

Seja $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ um conjunto de observações. O método K-Means é um processo iterativo que resulta na partição de X em K agrupamentos $P_1, \ldots, P_K$, e seus respectivos centroides $\boldsymbol{g}_k$, $k = 1, \ldots, K$, através da minimização da seguinte

função objetivo:

$$J = \sum_{k=1}^{K} \sum_{\boldsymbol{x}_i \in P_k} (\boldsymbol{x}_i - \boldsymbol{g}_k)'(\boldsymbol{x}_i - \boldsymbol{g}_k) = \sum_{k=1}^{K} \sum_{x_i \in P_k} \sum_{j=1}^{p} (x_{ij} - g_{kj})^2. \tag{2.9}$$

O algoritmo se inicia a partir de uma partição aleatória e alterna entre as etapas de representação, na qual os centroides dos grupos são atualizados, e alocação, na qual as observações são realocadas aos grupos definindo a partição, de modo a minimizar a função objetivo dada pela Equação (2.9). A cada iteração, na etapa de representação a partição é mantida fixa e os centroides $\boldsymbol{g}_k$, $k = 1, \ldots, K$, são atualizados de acordo com:

$$\boldsymbol{g}_k = \frac{1}{|P_k|} \sum_{\boldsymbol{x}_i \in P_k} \boldsymbol{x}_i, \tag{2.10}$$

em que $|P_k|$ é a cardinalidade de P_k, $k = 1, \ldots, K$. Na etapa de alocação os centroides são mantidos fixos e as partições P_k, $k = 1, \ldots, K$, são atualizadas de acordo com:

$$P_k = \{\boldsymbol{x}_i \in X : (\boldsymbol{x}_i - \boldsymbol{g}_k)'(\boldsymbol{x}_i - \boldsymbol{g}_k) \leq (\boldsymbol{x}_i - \boldsymbol{g}_h)'(\boldsymbol{x}_i - \boldsymbol{g}_h), \forall h \neq k, h = 1, \ldots, K\}. \tag{2.11}$$

Logo, o K-Means se resume ao seguinte algoritmo baseado nas etapas anteriores:

1. Inicialização
   Fixe o número de agrupamentos K ($2 \leq K < n$). Escolha aleatoriamente uma partição inicial P de Ω em K grupos $P_1, \ldots, P_K$ ou, alternativamente, escolha aleatoriamente K observações $\boldsymbol{g}_1, \ldots, \boldsymbol{g}_K$ pertencendo a Ω como protótipos iniciais e aloque cada observação i de acordo com o protótipo mais próximo $\boldsymbol{g}_h$ ($h = \arg\min_{1 \leq k \leq K} ||\boldsymbol{x}_i - \boldsymbol{g}_k||^2$) para obter a partição inicial $P = \{P_1, \ldots, P_K\}$

2. Atualização dos protótipos dos grupos
   Serão atualizados os protótipos dos grupos $\boldsymbol{g}_k$ ($k = 1, \ldots, K$), conforme a Equação (2.10);

3. Atualização da partição
   $test \leftarrow 0$
   para $i = 1$ até n faça
   defina o grupo vencedor P_h tal que

   $$h = \arg\min_{1 \leq k \leq K} ||\boldsymbol{x}_i - \boldsymbol{g}_k||^2$$

   se $i \in P_k$ e $h \neq k$

   $test \leftarrow 1$

   $P_h \leftarrow P_h \cup \{i\}$

$P_k \leftarrow P_k \setminus \{i\}$.

4. Critério de parada
   Se $test = 0$, então pare, caso contrário, volte ao passo (2).

Como o K-Means se trata de um algoritmo que possui dois passos, sendo eles representação e alocação, quando estamos na etapa de representação, a partição se mantém fixa e então encontramos os centroides. Já na etapa de alocação, são os centroides que são mantidos fixos e atualizamos as partições. Essas etapas se alternam até a convergência, quando nenhuma observação é realocada entre os grupos.

2.3 Agrupamento de séries temporais

Nas seções anteriores, apresentamos alguns conceitos a respeito de Séries Temporais e Métodos de Agrupamento. Sendo assim, podemos agrupar as séries temporais utilizando os métodos citados, porém adaptando-os às características delas.

Apesar das séries temporais consistirem em um fenômeno que se desenvolve ao longo do tempo, elas podem ser vistas como um objeto único de análise (Ramancharla e P.Nagabhushan, 2006) e, agrupar objetos com tal nível de complexidade pode trazer vantagens e a descoberta de padrões interessantes. Tais padrões podem ser comuns ou raros, por isso o agrupamento de séries temporais necessita de adaptações que levem em consideração suas características, bem como anomalias ou detectar *outliers* (Aghabozorgi, Shirkhorshidi, e Wah, 2015).

Descobrir padrões nos conjuntos de séries temporais é um dos principais objetivos do agrupamento delas. Além disso, agrupar séries temporais pode ajudar a descobrir anomalias, novidades ou detectar dissonâncias.

O agrupamento pode ajudar a reconhecer mudanças na dinâmica das séries, correlação entre elas, ajudar na predição de informações, entre outras utilidades. É comum aplicar o agrupamento de séries temporais em problemas nas áreas de aviação (Rebbapragada, Protopapas, Brodley, e Alcock, 2009), biologia (Subhani, Rueda, Ngom, e Burden, 2010), meio ambiente (Steinbach, Tan, Kumar, Klooster, e Potter, 2003), finanças (Kumar, Patel, e Woo, 2002), medicina (van den Heuvel, Mandl, e Pol, 2008), análise visual para tomada de decisão (Ali, Alqahtani, Jones, e Xie, 2019), hidrologia (Javed, Hamshaw, Rizzo, e Lee, 2019) entre outras. Se há dados dispostos em função do tempo, é possível agrupar séries temporais.

2.3.1 Formulação do problema

O agrupamento de séries temporais pode ser visto como uma etapa de preprocessamento no processo de mineração de dados ou como uma técnica de análise

isolada. É comum agrupar séries temporais quando se procura indexar, classificar ou detectar anomalias nas séries, como cita Abraham (2009). Neste trabalho foram utilizados apenas agrupamentos *hard*.

Quando agrupamos dados, utilizamos medidas de similaridades entre eles, e isso não é diferente com as séries temporais. Quando agrupamos as séries temporais, conseguimos identificar padrões interessantes e explorá-los.

O problema de agrupamento de séries temporais é definido, segundo Aghabozorgi, Shirkhorshidi, e Wah (2015), por: seja um conjunto de n séries temporais D $= \{F_1, F_2, \ldots, F_n\}$, o processo de partição de D em C $= \{C_1, C_2, \ldots, C_k\}$, de tal maneira que séries temporais homogêneas são agrupadas baseadas em certa medida de similaridade, é chamado de agrupamento de séries temporais. Então, C_i é chamado de cluster ou agrupamento, onde D $= U_{i=1}^{k}$ C_i e $C_i \cap C_j = \varnothing$ para i $\neq$ j.

2.3.2 Medidas de dissimilaridade no agrupamento de séries temporais

Algumas medidas de distâncias são compatíveis com representações de séries temporais. Quando se trata de séries temporais, as distâncias são aproximadas e não exatas, principalmente, quando se trata de séries com tamanhos amostrais diferentes.

Há várias medidas de distâncias utilizadas no agrupamento de séries temporais: distância Hausdorff, Haursdorff modificada (MODH), sincronização temporal dinâmica (DTW, sigla em inglês), distância baseada em Modelos Ocultos de Markov (HMM) e distância euclidiana (Hautamaki, Nykanen, e Franti, 2008) são as distâncias mais comuns.

A forma mais utilizada de calcular a distância entre séries temporais é considerá-las como séries temporais univariadas e calcular as distâncias, ponto a ponto, no decorrer da linha temporal.

Definição de distância entre séries temporais

Encontramos em Aghabozorgi, Shirkhorshidi, e Wah (2015) a seguinte definição para distância de séries temporais: seja $F_i = \{f_{i1}, ..., f_{it}, ..., f_{iT}\}$ uma série temporal de tamanho T, a distância entre duas séries temporais é definida no decorrer do tempo em todos os pontos, então $dist(F_i, F_j)$ será a soma das distâncias entre pontos individuais definida por

$$dist(F_i, F_j) = \sum_{t=1}^{T} dist(f_{it}, f_{jt}). \tag{2.12}$$

A escolha da distância apropriada depende da característica da série temporal, do seu tamanho, do método de representação e do objetivo do agrupamento, que pode ser:

Encontrar séries temporais similares no tempo

Neste objetivo, é comum utilizar a distância Euclidiana ou a correlação baseada nas distâncias. Porém, esse tipo de distância pode ser mais custosa computacionalmente. Assim, podem ser usadas séries temporais transformadas via transformação de Fourier, ondaletas ou aproximação agregada por partes.

Encontrar séries temporais similares na forma

Métodos elásticos como a sincronização temporal dinâmica são utilizados para calcular dissimilaridade. A similaridade na forma tem uma abrangência maior que a similaridade no tempo.

Encontrar séries temporais similares na mudança

Normalmente, para essa abordagem, são utilizados os modelos de Hidden Markov ou um processo ARMA, proposto por Box (1970), e a similaridade é medida nos parâmetros do modelo ajustado para séries temporais.

2.3.3 Algoritmos de agrupamento de séries temporais

Estudos anteriores, entre eles Warren Liao (2005), Javed, Lee, e Rizzo (2020) classificam os algoritmos de agrupamento de séries temporais em seis tipos, sendo eles:

1. Métodos de partição ou não-hierárquicos: define-se previamente o número de clusters que serão formados. O método de partição mais conhecido é o K-Means. Também podemos citar o K-medoid que tem basicamente o mesmo algoritmo do K-Means, porém utiliza o medoide como centroide.

2. Métodos hierárquicos: são os métodos aglomerativos ou divisivos. O primeiro considera, inicialmente, cada item como um cluster e agrupa-se até que haja apenas um cluster no final. No segundo, inicialmente, todos os itens pertencem a um grupo e divide-se até que haja tantos clusters quanto itens.

3. Baseado na grade: esse método baseia em uma espécie de grade que possui várias células, sendo um método normalmente menos utilizado no agrupamento de séries temporais.

4. Baseado no modelo: o principal objetivo é encontrar o modelo original do conjunto de dados. Ele considera cada agrupamento como um modelo diferente e ajusta os dados para aquele modelo. As principais desvantagens deste método é a necessidade de estabelecer os parâmetros antecipadamente, bem como suposições que podem não ser adequadas e, com isso, refletir em um insucesso do método. Além disso, o método é computacionalmente custoso.

5. Agrupamento baseado na densidade: devido a sua alta complexidade, este método não tem sido muito utilizado. Ele toma por base a densidade dos clusters e seus vizinhos. Imagine pontos em um espaço, os subespaços formados pelos pontos mais próximos formam um cluster.

6. Agrupamento multi-passos: esse método é uma forma prática de resumir todos os métodos que são híbridos, ou seja, utilizam passos de outros métodos para chegar ao resultado final. Seja um passo do método de partição e outro do método hierárquico, ou parte do algoritmo é baseado na grade e parte baseado no modelo, e assim por diante. Tudo para reduzir ao máximo o tempo e melhorar a qualidade do agrupamento.

2.3.4 Medidas de avaliação de agrupamento de séries temporais

Avaliar o agrupamento de séries temporais ainda é um problema em aberto e depende de fatores como a escolha do número de clusters, da medida de dissimilaridade, bem como fatores associados as séries temporais como seu domínio.

Existem várias métricas para avaliar o agrupamento de séries temporais. Técnicas de visualização e medidas escalares são as mais utilizadas para avaliação da qualidade dos agrupamentos. Nas medidas escalares, um número real único é gerado para representar a precisão de diferentes métodos de agrupamento. Estas medidas numéricas são classificadas em dois tipos:

- Índices externos: método de avaliação mais utilizado, conhecido também por validação externa, métodos extrínsecos, critério externo e método supervisionado. Nele, a solução é conhecida e o resultado obtido é comparado com essa solução;

- Índices internos: verificam a qualidade do método internamente, sem a informação sobre a solução ou nenhuma outra entrada externa.

Entre os índices externos existentes na literatura, podemos citar:

- Medida de similaridade de agrupamento (CSM, do inglês *Cluster Similarity Measure*); (WARREN LIAO, 2005)

- Índice Folkes e Mallow (ZHANG et al., 2006);
- Jaccard score (FOWLKES e MALLOWS, 2012);
- Índice Rand (RI, do inglês *Rand Index*) (CHIŞ et al., 2009);
- Índice Rand Ajustado (ARI, do inglês *Adjusted Rand Index*) (SANTOS e EMBRECHTS, 2009);
- F-measure (CHIŞ et al., 2009).

No estudo com dados simulados, como é de conhecimento a verdadeira partição das séries e o número de grupos, utilizamos o Índice de Rand Ajustado para avaliar e comparar a performance dos métodos de agrupamento. O Índice de Rand Ajustado é uma melhoria do Índice Rand, já que não é sensível ao número de classes nas partições ou à distribuição das observações nos grupos, e avalia o grau de concordância (similaridade) entre uma partição *a priori* e uma partição fornecida por um método de agrupamento. O Índice de Rand Ajustado assume valores no intervalo $[-1, 1]$, no qual o valor 1 indica concordância perfeita entre as partições, enquanto que valores próximos de zero ou negativos correspondem a concordância entre partições encontrada ao acaso (Chiş, Banerjee, e Hassanien, 2009).

Em situações práticas, como é o caso da utilização dos dados reais, não conhecemos as classes *a priori*. Neste caso, devemos utilizar algum índice interno para avaliar a qualidade dos agrupamentos. Ao contrário dos índices externos, como o Índice de Rand Ajustado, os índices internos não utilizam nenhum tipo de informação externa (como a partição *a priori*), se baseando apenas em informações presentes nos dados. Neste trabalho, utilizaremos os índices de Silhueta, Dunn e Conectividade para avaliar e comparar métodos de agrupamento aplicados a séries temporais intervalares oriundas de problemas reais. A Silhueta é uma medida do quão similar ao grupo ao qual pertence uma observação é, em comparação com outros grupos, assumindo valores no intervalo $[-1, 1]$. Observações bem agrupadas apresentando valores próximos de 1 e observações mal agrupadas apresentando valores próximos de -1 (Rousseeuw, 1987). O índice Dunn assume apenas valores positivos e quanto maior, melhor agrupados estão as observações. Já o índice Conectividade, apesar de também assumir valores positivos, o melhor agrupamento se dá quão mais perto de zero o índice estiver.

2.4 Agrupamento de séries temporais tipo-intervalo

Até aqui, revisamos os temas de séries temporais, métodos de agrupamentos para dados simbólicos e métodos de agrupamentos de séries temporais. Porém, ainda não

tratamos do agrupamento de séries temporais tipo-intervalo. Nesta seção, este tema será apresentado e, com isso, entraremos no assunto principal desta dissertação.

2.4.1 Variáveis tipo-intervalo

A variável tipo-intervalo é aquela representada por um intervalo. Seja Y uma variável tipo-intervalo, temos que Y = $\Im$ = [a,b] $\in \mathbb{R}$, com a $\leq$ b e a,b $\in \mathbb{R}$. Sendo que o intervalo pode ser aberto ou fechado em uma ou nas duas extremidades (Billard e Diday, 2006).

2.4.2 Séries temporais tipo-intervalo

A principal diferença entre uma série temporal simples e uma série temporal tipo-intervalo, também chamada de série temporal intervalar, ao invés da informação registrada representar um ponto, haverá um intervalo.

Um exemplo é a variação do câmbio no decorrer dos dias. O valor do dólar, por exemplo, pode ser melhor representado se citarmos o valor mínimo e o valor máximo dentro de um dia, ao invés de escolher apenas um destes valores ou analisar o valor de fechamento no dia.

Entre os métodos mais utilizados para o ajuste de séries temporais tipo-intervalo, temos o ARIMA univariado e multivariado, métodos que utilizam o centro e o raio como intervalos de previsão e também aqueles que utilizam os limites inferior e superior (Maia, Carvalho, e Ludermir, 2008). Também são utilizados os modelos de redes neurais multicamadas e funções de autocovariância e autocorrelação.

Para a previsão, os modelos de vetores autorregressivos e vetor de correção de erro, bem como filtros de suavização são os mais utilizados. Os modelos de intervalos autorregressivos para dados de séries temporais intervalares são também citados por Arroyo (2008), García e Maté (2010) e Rivera e Arroyo (2012). Em alguns casos, vemos, inclusive, a existência da correlação ou dependência contemporânea entre os limites inferiores e superiores (ou entre centro e raio) do intervalo.

Maharaj, Teles, e Brito (2019) definem uma série temporal tipo-intervalo como uma sequência de intervalos observados em sucessivos instantes de tempo. Cada um destes intervalos é representado pelos seus limites inferiores e superiores $[X_t] = [X_{t,L}; X_{t,U}], -\infty < X_{t,L} < X_{t,U} < \infty$, onde $X_{t,L}$ é o limite inferior e $X_{t,U}$ é o limite superior do intervalo. Também é possível a representação através do raio e do centro, por exemplo $[X_t] =< X_{t,C}; X_{t,R} >$, onde $X_{t,C} = (X_{t,L} + X_{t,U})/2$ e $X_{t,R} = (X_{t,U} - X_{t,L})/2$.

Apesar de estudos voltados para o tema de séries temporais tipo-intervalo, há pouca literatura referente ao agrupamento de séries temporais tipo-intervalo, o que demonstra a necessidade de mais estudos nesta área.

2.4.3 Distâncias para dados tipo-intervalo

É importante mencionar que a comparação entre séries temporais tipo-intervalo é possível se os pontos a serem comparados estiverem no mesmo instante de tempo. Além disso, os limites superior e inferior precisam se referir ao mesmo ponto. Para cada par de séries, a distância entre os intervalos observados é calculada em cada instante de tempo t = 1, 2, ..., T. Em seguida, a média das distâncias sobre todos os intervalos da série é calculada para obtermos o valor final da distância, sendo definida por:

$$D(\{[X_t]\}_i, \{[X_t]\}_j) = \frac{1}{T}\sum_{t=1}^{T} d([X_t]_i, [X_t]_j). \tag{2.13}$$

A distância Hausdorff é utilizada para agrupamento de séries temporais tipo-intervalo, ela é a distância máxima do conjunto para o ponto mais próximo do outro conjunto. A partir de dois intervalos $X_i = [X_{iL}, X_{iU}]$ e $X_j = [X_{jL}, X_{jU}]$ podemos defini-la como:

$$d_H = (X_i, X_j) = max\left\{|X_{iL} - X_{jL}|, |X_{iU} - X_{jU}|\right\}. \tag{2.14}$$

Outra métrica bastante utilizada no agrupamento de séries temporais tipo-intervalo é a distância de Minkowski:

$$d_k(X_i, X_j) = \sqrt[k]{|X_{iL} - X_{jL}|^k + |X_{iU} - X_{jU}|^k}. \tag{2.15}$$

Observa-se que, quando k = 2 na distância de Minskowski, teremos a distância Euclidiana:

$$d_2(X_i, X_j) = \sqrt{|X_{iL} - X_{jL}|^2 + |X_{iU} - X_{jU}|^2}. \tag{2.16}$$

Tais métricas serão importantes para obtermos a matriz de distâncias entre cada par de séries temporais e, a partir dessa matriz, utilizar o método de agrupamento escolhido para agrupar as séries temporais intervalares. Outras métricas podem ser obtidas em Chavent e Lechevallier (2002), Carvalho, Brito, e Bock (2006), Arroyo e Maté (2009) entre outros.

2.4.4 Agrupamento de séries temporais tipo-intervalo utilizando o método K-Means

O agrupamento de séries temporais de dados tipo intervalo realizado em Maharaj, Teles, e Brito (2019) utiliza o método K-Means (Macqueen, 1967) não nas séries originais, mas sim em transformações destas séries. Estas transformações são métodos de extrações de características ou parâmetros através de modelos gerando

matrizes de dados nas quais os métodos de agrupamentos serão aplicados. Maharaj, Teles, e Brito (2019) utiliza 4 formas de construir essas matrizes de dados a saber:

Utilizando a matriz de características

Dado que uma série temporal tipo-intervalo pode ser representada em termos do centro e raio, Maharaj, Teles, e Brito propõem utilizar as características do centro e raio destas séries como variáveis a serem agrupadas. As características utilizadas são: média, a variância, a assimetria, a curtose e a tendência do centro e raio de cada uma das séries. Tais características são dispostas em uma matriz, onde aplicamos um método de agrupamento. Utilizaremos a sigla FM (do inglês *Feature Matrix*) para nos referirmos a esta abordagem.

Utilizando estimativas dos parâmetros de modelos espaço-tempo

Em Maharaj, Teles, e Brito (2019), encontramos que os modelos autorregressivos espaço-tempo generalizam os modelos autorregressivos de séries temporais pela incorporação da dimensão espacial e são, com isso, caracterizados pela dependência linear no espaço e no tempo. Os modelos autorregressivos espaço-tempo são conhecidos pela sigla em inglês STAR(p), onde p representa a ordem autorregressiva.

Esse modelo pode ser utilizado para o ajuste de séries temporais tipo-intervalo. Seja a série temporal intervalar $[X_t] = [X_{t,L}; X_{t,U}]$, definimos o modelo STAR(p) como:

$$X_{t,L} = \theta_L + \sum_{i=1}^{p} \phi_i X_{t-i,L} + \sum_{i=0}^{p} \psi_i X_{t-i,U} + a_{t,L}$$

$$X_{t,U} = \theta_U + \sum_{i=1}^{p} \phi_i X_{t-i,U} + \sum_{i=0}^{p} \psi_i X_{t-i,L} + a_{t,U}, \tag{2.17}$$

onde θ_L e θ_U são os interceptos, ϕ_i (i = 1, 2, ..., p) e ψ_i (j = 0, 1, ..., p) são os parâmetros autorregressivos e $a_{t,L}$ e $a_{t,U}$ são processos de ruído branco com média zero, variâncias σ^2_{aL} e σ^2_{aU}, respectivamente, e covariância $\text{Cov}(a_{t,L}, a_{t,U}) = \sigma_{aL}$ com $\text{Cov}(a_{t,L}, a_{t+k,U}) = 0$ se k $\neq$ 0.

Estes parâmetros podem ser estimados por mínimos quadrados de três estágios ou máxima verossimilhança, assumindo normalidade. Dado o conjunto de séries temporais de dado tipo intervalo que nós queremos agrupar, Maharaj, Teles, e Brito (2019) propõem usar as estimativas de parâmetros do modelo STAR ajustado como variáveis de agrupamento para as quais uma métrica Euclidiana é determinada. Na matriz gerada, os métodos de agrupamento são aplicados.

Utilizando matrizes de funções de autocorrelação

Maharaj, Teles, e Brito (2019) propõem este método de agrupamento de séries temporais intervalares para o caso de séries bivariadas e sua matriz de função de autocorrelação pode ser definida como:

$$\rho(k) = \begin{bmatrix} \rho_{LL}(k) & \rho_{LU}(k) \\ \rho_{UL}(k) & \rho_{UU}(k) \end{bmatrix}, k = 1, 2, \ldots, K,$$

em que $\rho_{LL}(k)$ e $\rho_{UU}(k)$ são as funções de autocorrelações de $X_{t,L}$ e $X_{t,U}$ respectivamente, e $\rho_{LU}(k)$ e $\rho_{UL}(k)$ são funções de correlação cruzada. Sejam as matrizes

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} e \quad B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}.$$

As seguintes medidas de distância entre elas podem ser utilizadas:

- Distância Frobenius: para qualquer matriz 2x2 a distância entre A e B pode ser calculada por

$$\sqrt{\sum_{i=1}^{2}\sum_{j=1}^{2}(a_{ij} - b_{ij})^2}.$$

  Logo, tal distância é aplicada a matriz de função de autocorrelação de quaisquer pares de séries temporais X1 e X2:

$$\left\{ \sum_{k=1}^{K} \Big[[\rho_{LL_{X1}}(k) - \rho_{LL_{X2}}(k)]^2 + [\rho_{LU_{X1}}(k) - \rho_{LU_{X2}}(k)]^2 + [\rho_{UL_{X1}}(k) - \rho_{UL_{X2}}(k)]^2 + [\rho_{UU_{X1}}(k) - \rho_{UU_{X2}}(k)]^2 \Big] \right\}^{1/2}.$$

- Valor Absoluto: a medida de distância entre as matrizes A e B é dada por

$$\sum_{i=1}^{2}\sum_{j=1}^{2}|a_{ij} - b_{ij}|.$$

  Assim, a distância é aplicada a matriz de funções de autocorrelação de X1 e

X2:

$$\sum_{k=1}^{K} \Big[|\rho_{LL_{X1}}(k) - \rho_{LL_{X2}}(k)| + |\rho_{LU_{X1}}(k) - \rho_{LU_{X2}}(k)| + |\rho_{UL_{X1}}(k) - \rho_{UL_{X2}}(k)| + |\rho_{UU_{X1}}(k) - \rho_{UU_{X2}}(k)| \Big].$$

- Norma da máxima soma absoluta das linhas: a medida de distância entre A e B será

$$\max_i \sum_j |a_{ij} - b_{ij}| (i, j = 1, 2)$$

e a distância entre a matriz de função de autocorrelação de X1 e X2:

$$\sum_{k=1}^{K} [max\{|\rho_{LL_{X1}}(k) - \rho_{LL_{X2}}(k)| + |\rho_{LU_{X1}}(k) - \rho_{LU_{X2}}(k)|, |\rho_{UL_{X1}}(k) - \rho_{UL_{X2}}(k)| + |\rho_{UU_{X1}}(k) - \rho_{UU_{X2}}(k)|\}].$$

- Norma da máxima soma absoluta das colunas: a medida de distância entre A e B será

$$\max_j \sum_i |a_{ij} - b_{ij}| (i, j = 1, 2)$$

levando a

$$\sum_{k=1}^{K} [max\{|\rho_{LL_{X1}}(k) - \rho_{LL_{X2}}(k)| + |\rho_{UL_{X1}}(k) - \rho_{UL_{X2}}(k)|, |\rho_{LU_{X1}}(k) - \rho_{LU_{X2}}(k)| + |\rho_{UU_{X1}}(k) - \rho_{UU_{X2}}(k)|\}].$$

As funções de autocorrelação e correlação cruzada das matrizes deverão ser estimadas.

Utilizando funções de autocorrelação intervalar

Também se propõe agrupar com distâncias baseadas na função de autocorrelação da amostra especificamente definido para uma série temporal intervalar. A proposta é utilizar a média e a covariância da mesma amostra baseada na decomposição da variabilidade entre intervalos. Esta abordagem leva a melhorar funções de auto-covariância e autocorrelação das amostras de séries temporais de dados tipo intervalo.

Os momentos amostrais utilizados a seguir assumem estacionariedade fraca e distribuição uniforme do intervalo dado.

Média amostral:

$$\overline{X} = \frac{1}{2n}\sum_{t=1}^{n}(X_{t,L} + X_{t,U}) = \frac{1}{n}\sum_{t=1}^{n} X_{t,C}. \tag{2.18}$$

Variância amostral:

$$\hat{\gamma}_0 = \frac{1}{3n}\sum_{t=1}^{n}(X_{t,L}^2 + X_{t,L}X_{t,U} + X_{t,U}^2) - \overline{X}^2. \tag{2.19}$$

Auto-covariância amostral:

$$\begin{aligned}\hat{\gamma}_k = \tfrac{1}{6n}\sum_{t=1}^{n-k}[&2(X_{t,L} - \overline{X})(X_{t+k,L} - \overline{X})\\ &+(X_{t,L} - \overline{X})(X_{t+k,U} - \overline{X})\\ &+(X_{t,U} - \overline{X})(X_{t+k,L} - \overline{X})\\ &+2(X_{t,U} - \overline{X})(X_{t+k,U} - \overline{X})].\end{aligned} \tag{2.20}$$

Podemos observar que, se substituirmos k por 0(zero) na equação (2.20), chegaremos a equação (2.19). A função de autocorrelação pode ser definida como $\hat{\rho}_k = \hat{\gamma}_k/\hat{\gamma}_0$. Dado o conjunto de séries temporais de dados tipo intervalo que nós queremos agrupar, nós utilizaremos a autocorrelação da amostra para um número específico de defasagens como a variável de agrupamento para obter a distância euclidiana. Estas soluções de agrupamento podem ser utilizadas em métodos hierárquicos e não hierárquicos.

2.5 Funções Kernel

As funções Kernel aplicadas nos métodos de agrupamentos torna possível a aplicação deste métodos a conjuntos de dados não linearmente separáveis. (Filippone, Camastra, Masulli, e Rovetta, 2008)

Seja $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ um conjunto não-vazio de observações no espaço p-dimensional $\mathbb{R}^p$. Uma função $\mathcal{K}$: $X \times X \to \mathbb{R}$ é dito ser um *kernel* positivo-definido se, e somente se, $\mathcal{K}$ for simétrica, ou seja, $\mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \mathcal{K}(\boldsymbol{x}_j, \boldsymbol{x}_i)$ e (Mercer, 1909):

$$\sum_{i=1}^{n}\sum_{j=1}^{n} c_i c_j \mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq 0, \forall\, n \geq 2. \tag{2.21}$$

Considere $\Phi : X \to \mathcal{F}$ uma transformação não-linear qualquer do espaço de entrada de X para um espaço de mais alta dimensão $\mathcal{F}$. Ao aplicar a transformação

Φ, o produto interno $\boldsymbol{x}_i^\top \boldsymbol{x}_j$ é mapeado no espaço de característica como $\Phi(\boldsymbol{x}_i)^\top \Phi(\boldsymbol{x}_j)$. O fundamento principal do uso de *kernel* se deve ao fato de que o mapeamento Φ não precisa ser explicitamente especificado, por causa de que todo *kernel* que satisfaz as condições anteriores pode ser escrito como:

$$\mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \Phi(\boldsymbol{x}_i)^\top \Phi(\boldsymbol{x}_j) \tag{2.22}$$

que é conhecida como *kernel trick* (Schölkopf, Smola, e Müller, 1998).

Sendo assim, a distância euclidiana, por exemplo, pode ser escrita da seguinte forma:

$$\begin{aligned} ||\Phi(\boldsymbol{x}_i) - \Phi(\boldsymbol{x}_j)||^2 &= [\Phi(\boldsymbol{x}_i) - \Phi(\boldsymbol{x}_j)]^\top [\Phi \boldsymbol{x}_i) - \Phi(\boldsymbol{x}_j)] \\ &= \Phi(\boldsymbol{x}_i)^\top \Phi(\boldsymbol{x}_j) - 2\Phi(\boldsymbol{x}_i)^\top \Phi(\boldsymbol{x}_j)] + \Phi(\boldsymbol{x}_j)^\top \Phi(\boldsymbol{x}_i) \\ &= \mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) - 2\mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) + \mathcal{K}(\boldsymbol{x}_j, \boldsymbol{x}_i). \end{aligned}$$

As funções *kernel* mais usuais na literatura são:

- Gaussiano: $\mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) = exp\Big(\frac{-||\boldsymbol{x}_i - \boldsymbol{x}_j||^2}{2\sigma^2}\Big)$, para $\sigma > 0$;
- Polinomial: $\mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\lambda \boldsymbol{x}_i^T \boldsymbol{x}_j + \theta)^d$, para $\lambda > 0, \theta \geq 0, d \in \mathbb{N}$;
- Linear: $\mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i^\top \boldsymbol{x}_j$;
- Laplaciano: $\mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) = exp(-\lambda||\boldsymbol{x}_i - \boldsymbol{x}_j$, para $\lambda > 0$;
- Sigmoidal: $\mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) = tanh(\lambda \boldsymbol{x}_i^T \boldsymbol{x}_j + \theta)$, para $\lambda > 0, \theta \geq 0$.

No presente trabalho utilizaremos o Kernel Gaussiano, que é amplamente utilizado na literatura e possui suas facilidades matemáticas.

2.5.1 Funções de *kernel* para intervalos

O Kernel Gaussiano ou função RBF é comumente utilizado para combater o problema da não linearidade no espaço original (Costa, Pimentel, e de Souza, 2010). É possível aplicar a kernelização em uma componente, em que a expressão é representada da seguinte forma:

$$\mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_k) = \exp\Big(\frac{-||\boldsymbol{x}_i - \boldsymbol{x}_k||^2}{2\sigma^2}\Big), \; para \quad \sigma > 0, \tag{2.23}$$

em que

$$||\boldsymbol{x}_i - \boldsymbol{x}_k||^2 = \sum_{j=1}^{p} [(x_{iL_j} - x_{kL_j})^2 + (x_{iU_j} - x_{kU_j})^2], \tag{2.24}$$

onde $\boldsymbol{x}_i = ([x_{iL_1}, x_{iU_1}], \ldots, [x_{iL_p}, x_{iU_p}])$ e $\boldsymbol{x}_k = ([x_{kL_1}, x_{kU_1}], \ldots, [x_{kL_p}, x_{kU_p}])$ são os vetores $2p$-dimensionais representando o i-ésimo e o k-ésimo objetos do conjunto $\Omega = 1, \ldots, n$ que é descrito por p variáveis simbólicas tipo intervalo.

Quando a kernelização é aplicada a duas componentes, a função é representada da seguinte forma:

$$\mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\Big(\frac{-||\boldsymbol{x}_{iL} - \boldsymbol{x}_{kL}||^2}{2\sigma^2}\Big) + \exp\Big(\frac{-||\boldsymbol{x}_{iU} - \boldsymbol{x}_{kU}||^2}{2\sigma^2}\Big), \quad \sigma > 0, \tag{2.25}$$

em que

$$||\boldsymbol{x}_{iL} - \boldsymbol{x}_{kL}||^2 = \sum_{j=1}^{p}(x_{iL_j} - x_{kL_j})^2 \quad \text{e} \quad ||\boldsymbol{x}_{iU} - \boldsymbol{x}_{kU}||^2 = \sum_{j=1}^{p}(x_{iU_j} - x_{kU_j})^2, \tag{2.26}$$

onde $\boldsymbol{x}_{iL} = (x_{iL_1}, \ldots, x_{iL_p})$ e $\boldsymbol{x}_{iU} = (x_{iU_1}, \ldots, x_{iU_p})$ são os vetores p-dimensionais sendo respectivamente relacionados aos limites inferiores e superiores dos intervalos representando o i-ésimo do elemento de Ω e $\boldsymbol{x}_{kL} = (x_{kL_1}, \ldots, x_{kL_p})$ e $\boldsymbol{x}_{kU} = (x_{kU_1}, \ldots, x_{kU_p})$ representam o k-ésimo elemento de Ω.

2.6 Métodos de agrupamento para intervalos

Nesta seção apresentamos os métodos de agrupamento para dados tipo-intervalo considerados nessa dissertação.

2.6.1 Kernel K-Means baseado na Kernelização da métrica

Dado um conjunto Ω de intervalos, onde cada objeto está representado por $\boldsymbol{x}_i$ $(i = 1, \ldots, n)$, sendo $\boldsymbol{x}_i = ([x_{iL_1}, x_{iU_1}], \ldots, [x_{iL_p}, x_{iU_p}])^T$ e centroides $\boldsymbol{g}_k$ $(k = 1, \ldots, K)$, sendo $\boldsymbol{g}_k = ([\alpha_{k1}, \beta_{k1}], \ldots, [\alpha_{kp}, \beta_{kp}])^T$. Seja $P = \{P_1, \ldots, P_K\}$ uma partição de Ω em K grupos.

O algoritmo Kernel K-Means com kernelização da métrica para intervalos busca por um conjunto de centroides $\boldsymbol{g}_k$ $(k = 1, \ldots, K)$ e uma partição P que minimizam a seguinte função objetivo:

$$\begin{aligned} J^{(t)} &= \sum_{k=1}^{K}\sum_{i \in P_k^{(t)}} ||\Phi(\boldsymbol{x}_i) - \Phi(\boldsymbol{g}_k^{(t)})||^2 \\ &= \sum_{k=1}^{K}\sum_{i \in P_k^{(t)}} \mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_i) - 2\mathcal{K}(\boldsymbol{x}_i, \boldsymbol{g}_k^{(t)}) + \mathcal{K}(\boldsymbol{g}_k^{(t)}, \boldsymbol{g}_k^{(t)}). \end{aligned} \tag{2.27}$$

Nesse caso, escolhemos o *kernel* Gaussiano, então teremos as componentes α_{kj} e

β_{kj} dos centroides dos grupos $\boldsymbol{g}_k$ $(k = 1, \ldots, K)$ atualizadas conforme:

$$\alpha_{kj}^{(t+1)} = \frac{\sum\limits_{i \in P_k^{(t)}} \mathcal{K}(\boldsymbol{x}_i, \boldsymbol{g}_k^{(t)}) x_{iL_j}}{\sum\limits_{i \in P_k^{(t)}} \mathcal{K}(\boldsymbol{x}_i, \boldsymbol{g}_k^{(t)})} \tag{2.28}$$

e

$$\beta_{kj}^{(t+1)} = \frac{\sum\limits_{i \in P_k^{(t)}} \mathcal{K}(\boldsymbol{x}_i, \boldsymbol{g}_k^{(t)}) x_{iU_j}}{\sum\limits_{i \in P_k^{(t)}} \mathcal{K}(\boldsymbol{x}_i, \boldsymbol{g}_k^{(t)})}, \tag{2.29}$$

respectivamente, sendo t a t-ésima iteração do algoritmo e t+1 a iteração subsequente.

Após essa etapa, os centroides são mantidos fixos e a partição $P = P_1, \ldots, P_k$ que minimiza a função objetivo em (2.27) é atualizada de acordo com a seguinte regra de alocação:

$$P_k^{(t)} = \left\{ i \in \Omega : ||\Phi(\boldsymbol{x}_i) - \Phi(\boldsymbol{g}_k^{(t)})||^2 \leq ||(\Phi(\boldsymbol{x}_i) - \Phi(\boldsymbol{g}_h^{(t)})||^2, \forall h \neq k, h = 1, \ldots, K \right\} \tag{2.30}$$

Com isso, o algoritmo do Kernel K-Means com a kernelização da métrica fica com as seguintes adaptações:

1. Inicialização
   Fixe o número de agrupamentos K $(2 \leq K < n)$. Escolha aleatoriamente uma partição inicial P de Ω em K grupos $P_1, \ldots, P_K$ ou, alternativamente, escolha aleatoriamente K observações $\boldsymbol{g}_1, \ldots, \boldsymbol{g}_K$ pertencendo a Ω como protótipos iniciais e aloque cada observação i de acordo com o protótipo mais próximo $\boldsymbol{g}_h$ $(h = \arg\min_{1 \leq k \leq K} ||\Phi(\boldsymbol{x}_i) - \Phi(\boldsymbol{g}_k^{(t)})||^2)$ para obter a partição inicial $P = \{P_1, \ldots, P_K\}$;

2. Atualização dos protótipos dos grupos
   Serão atualizadas as componentes α_{kj} e β_{kj} nos protótipos dos grupos $\boldsymbol{g}_k$ $(k = 1, \ldots, K)$, conforme as equações (2.28) e (2.29).

3. Atualização da partição
   $test \leftarrow 0$
   para $i = 1$ até n faça

   defina o grupo vencedor P_h tal que

   $h = \arg\min_{1 \leq k \leq K} ||\Phi(\boldsymbol{g}_i) - \Phi(\boldsymbol{g}_k)||^2$

   se $i \in P_k$ e $h \neq k$

$test \leftarrow 1$

$P_h \leftarrow P_h \cup \{i\}$

$P_k \leftarrow P_k \setminus \{i\}$.

4. Critério de parada
   Se $test = 0$, então pare, caso contrário, volte ao passo (2).

2.6.2 Kernel K-Means no espaço de características

O espaço de características ao qual nos referimos na seção anterior se trata do espaço para o qual o espaço original das observações é mapeado através de uma transformação não-linear. (Kim, Lee, Lee, e Lee, 2005)

É importante citar que a transformação não-linear Φ não é conhecida, logo os protótipos no espaço de características não podem ser obtidos explicitamente da equação dos centroides. Contudo, a distância $||\Phi(\mathbf{x}_i) - \mathbf{g}_k^{\Phi}||^2$ pode ser obtida utilizando o *kernel trick*:

$$\begin{aligned} ||\Phi(\boldsymbol{x}_i) - (\boldsymbol{g}_j^{\Phi})||^2 &= \Phi(\boldsymbol{x}_i)^\top \Phi(\boldsymbol{x}_j) - 2\Phi(\boldsymbol{x}_i)^\top (\boldsymbol{g}_j^{\Phi}) + (\boldsymbol{g}_j^{\Phi})^\top (\boldsymbol{g}_j^{\Phi}) \\ &= \Phi(\boldsymbol{x}_i)^\top \Phi(\boldsymbol{x}_i) - 2\,\frac{\sum_{l \in P_k} \Phi(\boldsymbol{x}_l)^\top \Phi(\boldsymbol{x}_i)}{|P_k|} + \frac{\sum_{r \in P_k} \sum_{s \in P_k} \Phi(\boldsymbol{x}_r)^\top \Phi(\boldsymbol{x}_s)}{|P_k|^2} \\ &= \mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_i) - 2\,\frac{\sum_{l \in P_k} \mathcal{K}(\boldsymbol{x}_l, \boldsymbol{x}_i)}{|P_k|} + \frac{\sum_{r \in P_k} \sum_{s \in P_k} \mathcal{K}(\boldsymbol{x}_r, \boldsymbol{x}_s)}{|P_k|^2}. \end{aligned} \tag{2.31}$$

A principal diferença do Kernel K-Means no espaço de características para o Kernel K-Means baseado na kernelização da métrica é que, no primeiro caso, os centroides são obtidos no espaço de características e as distâncias das observações para os centroides nesse novo espaço são calculadas através de funções kernel, enquanto que, no segundo caso, os centroides são obtidos no espaço original dos dados e as distâncias são calculadas através de funções kernel.

$$(\mathbf{g}_k^{\Phi})^{(t+1)} = \frac{1}{|P_k^{(t)}|} \sum_{i \in P_k^{(t)}} \Phi(\boldsymbol{x}_i). \tag{2.32}$$

O algoritmo Kernel K-Means no espaço de características para intervalos busca por um conjunto de centroides $\boldsymbol{g}_k^{\Phi}$ $(k = 1, \ldots, K)$ e uma partição P que minimizam a seguinte função objetivo:

$$J^{(t)} = \sum_{k=1}^{K} \sum_{i \in P_k^{(t)}} \left\{ \mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_i) - 2\,\frac{\sum_{l \in P_k} \mathcal{K}(\boldsymbol{x}_l, \boldsymbol{x}_i)}{|P_k|} + \frac{\sum_{r \in P_k} \sum_{s \in P_k} \mathcal{K}(\boldsymbol{x}_r, \boldsymbol{x}_s)}{|P_k|^2} \right\}. \tag{2.33}$$

Note novamente que os centroides não podem ser explicitamente calculados porque não conhecemos o mapeamento $\Phi(\cdot)$. Contudo, as distâncias entre as observações e os centroides dos grupos podem ser calculadas (devido ao *kernel trick*) através da Equação (2.31). Nesse caso, as distâncias são recalculadas a cada passo de acordo com a partição P, que é atualizada de acordo com a seguinte regra:

$$P_k^{(t)} = \left\{ i \in \Omega : ||\Phi(\boldsymbol{x}_i) - (\boldsymbol{g}_k^{\Phi})^{(t)})||^2 \leq ||\Phi(\boldsymbol{x}_i) - (\boldsymbol{g}_h^{\Phi})^{(t)})||^2, \forall h \neq k, h = 1, \ldots, K \right\}. \tag{2.34}$$

Após todas essas adaptações, o algoritmo Kernel K-Means no espaço de características seguirá o passo a passo a seguir:

1. Inicialização
   Fixe o número de grupos K, $2 \leq K < n$. Neste caso, não é possível calcular os centroides dos grupos. Então a escolha da partição P será aleatória e não haverá etapa de atualização dos protótipos.

2. Atualização da partição

   $test \leftarrow 0$
   para $i = 1$ até n faça
   defina o grupo vencedor P_h tal que
   $h = arg\ \ min_{1 \leq k \leq K} ||\Phi(\boldsymbol{x}_i) - \Phi(\boldsymbol{g}_k^{(t)})||^2$
   se $i \in P_k$ e $h \neq k$
   $test \leftarrow 1$
   $P_h \leftarrow P_h \cup \{i\}$
   $P_k \leftarrow P_k \setminus \{i\}$.

3. Critério de parada
   Se $test = 0$, então pare, caso contrário, volte ao passo(2)

2.7 Agrupamento baseado em Distâncias Adaptativas

As distâncias adaptativas são distâncias calculadas baseadas em parâmetros que irão dar pesos maiores às variáveis mais importantes para o resultado. A vantagem disso é que, a cada iteração, o próprio algoritmo recalcula o parâmetro de acordo com os resultados preliminares que forem surgindo conforme tamanhos e formas diferentes. Aqui iremos aplicar estas distâncias adaptativas para dados tipo intervalo.

Assim como foi ajustado nas versões kernelizadas, as distâncias adaptativas geram alterações nas equações para os cálculos das distâncias e dos algoritmos.

A expressão (2.35) abaixo é uma distância adaptativa inspirada na distância euclidiana, em outras palavras podemos dizer que a expressão é a distância euclidiana adaptada.

Dado $\boldsymbol{\lambda}_k = (\lambda_{1k}, \ldots \lambda_{pk})$ um vetor de parâmetros da distância adaptativa,

$$d_k^2(\mathbf{x}_i, \mathbf{g}_k) = \sum_{j=1}^{p} \lambda_{jk} \Big[\Big(x_{iL_j} - \alpha_{kj} \Big)^2 + \Big(x_{iU_j} - \beta_{kj} \Big)^2 \Big], \tag{2.35}$$

em que $\lambda_{jk} > 0$ e $\prod_{j=1}^{p} \lambda_{jk} = 1$.

Há duas principais formas de incorporar pesos no agrupamento de dados do tipo intervalo. Uma delas é introduzindo os parâmetros de peso no termo do somatório das distâncias ao nível das variáveis. Como pode ser visto na Equação (2.35) a uma ponderação comum foi aplicada às parcelas referentes à soma dos quadrados das diferenças entre os limites inferiores e superiores do somatório, ou seja, os limites inferior e superior de uma variável intervalar recebem o mesmo peso. Sendo x_{iL_j} e α_{kj} os limites inferiores e x_{iU_j} e β_{kj} os limites superiores. Já na expressão abaixo, podemos observar que os parâmetros de ponderação são aplicados a cada diferença entre os limites inferiores e superiores, ou seja, os limites inferior e superior de uma mesma variável intervalar podem receber pesos diferentes:

$$d_k^2(\mathbf{x}_i, \mathbf{g}_k) = \sum_{j=1}^{p} \Big[\lambda_{jkL} (x_{iL_j} - \alpha_{kj})^2 + \lambda_{jkU} (x_{iU_j} - \beta_{kj})^2 \Big], \tag{2.36}$$

em que $\lambda_{jkL} > 0$ e $\prod_{j=1}^{p} \lambda_{jkL} = 1$ e $\lambda_{jkU} > 0$ e $\prod_{j=1}^{p} \lambda_{jkU} = 1$.

K-Means baseado em distâncias adaptativas

Considere novamente um conjunto Ω de intervalos, onde cada objeto está representado por $\boldsymbol{x}_i$ $(i = 1, \ldots, n)$, sendo $\boldsymbol{x}_i = ([x_{iL_1}, x_{iU_1}], \ldots, [x_{iL_p}, x_{iU_p}])^T$ e centroides $\boldsymbol{g}_k$ $(k = 1, \ldots, K)$, sendo $\boldsymbol{g}_k = ([\alpha_{k1}, \beta_{k1}], \ldots, [\alpha_{kp}, \beta_{kp}])^T$. Seja $P = P_1, \ldots, P_K$ uma partição de Ω em K grupos.

O algoritmo K-Means baseado em distâncias adaptativas para intervalos com vetor de pesos comum aos limites inferior e superior busca por um conjunto de centroides $\boldsymbol{g}_k$ $(k = 1, \ldots, K)$, vetores de pesos $\boldsymbol{\lambda}_k$ $(k = 1, \ldots, K)$ e uma partição P que minimizam a seguinte função objetivo:

$$
\begin{aligned}
J &= \sum_{k=1}^{K} \sum_{x_i \in P_k} \sum_{j=1}^{p} (\boldsymbol{x}_{ij} - \boldsymbol{g}_{kj})^2 \\
&= \sum_{k=1}^{K} \sum_{i \in P_k} \sum_{j=1}^{p} \lambda_{jk} \Big[\Big(x_{iL_j} - \alpha_{kj} \Big)^2 + \Big(x_{iU_j} - \beta_{kj} \Big)^2 \Big].
\end{aligned}
\tag{2.37}
$$

Nesse caso, estamos considerando apenas a Equação (2.35), na qual apenas um parâmetro é aplicado ao somatório dos quadrados das distâncias. Assim, teremos as componentes α_{kj} e β_{kj} dos centroides dos grupos $\mathbf{g}_k$ $(k = 1, \ldots, K)$ atualizadas de acordo com:

$$
\hat{\alpha}_{jk} = \frac{1}{n_k} \sum_{j=1} x_{jL_i} \tag{2.38}
$$

e

$$
\hat{\beta}_{jk} = \frac{1}{n_k} \sum_{j=1} x_{jU_i}, \tag{2.39}
$$

respectivamente, em que n_k é o cardinal da classe P_k.

Após essa etapa, os centroides são mantidos fixos, e o vetor de parâmetros $\boldsymbol{\lambda}_k = (\lambda_{1k}, \ldots \lambda_{pk})$, sob as restrições $\lambda_{pk} > 0$ e $\prod_{j=1}^{p} \lambda_{jk} = 1$, que minimize a função objetivo é atualizado de acordo com:

$$
\hat{\lambda}_{jk} = \frac{\prod_{h=1}^{p} \Big(\sum_{i \in P_k} (x_{iL_j} - \hat{\alpha}_{kj})^2 + (x_{iU_j} - \hat{\beta}_{kj})^2 \Big)^{1/p}}{\sum_{i \in P_k} (x_{iL_j} - \hat{\alpha}_{kj})^2 + (x_{iU_j} - \hat{\beta}_{kj})^2}. \tag{2.40}
$$

A partição $P = \{P_1, \ldots, P_k\}$ que minimiza a função objetivo é atualizada, mantidos os centroides e os vetores de pesos fixos, de acordo com a seguinte regra de alocação:

$$
P_k^{(t)} = \Big\{ i \in \Omega : d^2(\boldsymbol{x}_i, \boldsymbol{g}_k^{(t)}) \leq d^2(\boldsymbol{x}_i, \boldsymbol{g}_l^{(t)}), \forall\, l \neq k,\ l = 1, \ldots, K \Big\}. \tag{2.41}
$$

Com isso, o algoritmo do K-Means baseado em distâncias adaptativas fica com as seguintes modificações:

1. Inicialização

   Fixe o número de agrupamentos K $(2 \leq K < n)$. Faça $\lambda_{jk} = 1$ $\forall j \in \{1, 2, \ldots, p\}$ e $\forall k \in \{1, 2, \ldots, K\}$. Escolha aleatoriamente uma partição inicial P de Ω em K grupos $P_1, \ldots, P_K$ ou, alternativamente, escolha aleatoriamente K observações $\boldsymbol{g}_1, \ldots, \boldsymbol{g}_K$ pertencendo a Ω como protótipos iniciais e aloque cada observação i de acordo com o protótipo mais próximo $\boldsymbol{g}_h$

$(h = \arg\min_{1\leq k\leq K} \sum_{j=1}^{p} \lambda_{jk}[(x_{iL_j} - \alpha_{kj})^2 + (x_{iU_j} - \beta_{kj})^2])$ para obter a partição inicial $P = \{P_1, \ldots, P_K\}$;

2. Atualização dos protótipos
   Serão atualizadas as componentes α_{kj} e β_{kj} nos protótipos dos grupos $\mathbf{g}_k$ $(k = 1, \ldots, K)$, conforme as Equações (2.38) e (2.39).

3. Atualização das distâncias
   Os pesos serão calculados conforme a equação (2.40) e as distâncias atualizadas de acordo com a Equação (2.35).

4. Atualização da partição
   $test \leftarrow 0$
   para $i = 1$ até n faça

   defina o grupo vencedor P_h tal que

   $$h = \arg\min_{k=l,\ldots,K} = \sum_{j=1}^{p} \lambda_{jk}[(x_{iL_j} - \alpha_{kj})^2 + (x_{iU_j} - \beta_{kj})^2]$$

   se $i \in P_k$ e $h \neq k$

   $test \leftarrow 1$

   $P_h \leftarrow P_h \cup \{i\}$

   $P_k \leftarrow P_k \setminus \{i\}$.

5. Critério de parada
   Se $test = 0$, então pare, caso contrário, volte ao passo(2)

O algoritmo K-Means baseado em distâncias adaptativas para intervalos com vetor de pesos comum aos limites inferior e superior busca por um conjunto de centroides $\boldsymbol{g}_k$ $(k = 1, \ldots, K)$, vetores de pesos $\boldsymbol{\lambda}_{kI}$ e $\boldsymbol{\lambda}_{kS}$ $(k = 1, \ldots, K)$ e uma partição P que minimizam a seguinte função objetivo:

$$\begin{aligned} J &= \sum_{k=1}^{K} \sum_{x_i \in P_k} \sum_{j=1}^{p} (\boldsymbol{x}_{ij} - \boldsymbol{g}_{kj})^2 \\ &= \sum_{k=1}^{K} \sum_{x_i \in P_k} \sum_{j=1}^{p} \left[\lambda_{jkL}(x_{iL_j} - \alpha_{kj})^2 + \lambda_{jkU}(x_{iU_j} - \beta_{kj})^2\right]. \end{aligned} \tag{2.42}$$

As equações de atualização das componentes α_{kj} e β_{kj} dos centroides dos grupos $\boldsymbol{g}_k$ $(k = 1, \ldots, K)$ ficam as mesmas que para a equação com apenas um parâmetro, ou seja, a partição e os vetores de pesos são mantidos fixos e as componentes dos centroides são atualizadas de acordo com as Equações (2.38) e (2.39).

Na segunda etapa, com a partição e os centroides mantidos fixos, os vetores de parâmetros dos limites inferior $\boldsymbol{\lambda}_{kL} = (\lambda_{1kL}, \ldots \lambda_{pkL})$ e superior $\boldsymbol{\lambda}_{kU} = (\lambda_{1kU}, \ldots \lambda_{pkU})$ onde $\lambda_{jkL} > 0$ e $\lambda_{jkU} > 0$ $\forall j \in \{1, 2, \ldots, p\}$ e $\prod_{j=1}^{p} \lambda_{jkL} = 1$ e $\prod_{j=1}^{p} \lambda_{jkU} = 1$ $\forall k \in \{1, 2, \ldots, K\}$ são atualizados de acordo com:

$$\hat{\lambda}_{jkI} = \frac{\prod_{h=1}^{p} \Big(\sum_{i \in P_k} (x_{iL_h} - \hat{\alpha}_{kh})^2 \Big)^{1/p}}{\sum_{i \in P_k} (x_{iL_j} - \hat{\alpha}_{kj})^2} \tag{2.43}$$

e

$$\hat{\lambda}_{jkS} = \frac{\prod_{h=1}^{p} \Big(\sum_{i \in P_k} (x_{iU_h} - \hat{\beta}_{kh})^2 \Big)^{1/p}}{\sum_{i \in P_k} (x_{iU_j} - \hat{\beta}_{kj})^2}, \tag{2.44}$$

respectivamente.

A partição $P = \{P_1, \ldots, P_k\}$ que minimiza a função objetivo é atualizada, mantidos os centroides e os vetores de pesos fixos, de acordo com a seguinte regra de alocação:

$$P_k^{(t)} = \left\{ i \in \Omega : d^2(\boldsymbol{x}_i, \boldsymbol{g}_k^{(t)}) \leq d^2(\boldsymbol{x}_i, \boldsymbol{g}_l^{(t)}), \forall\, l \neq k,\ l = 1, \ldots, K \right\}. \tag{2.45}$$

Com isso, o algoritmo do K-Means baseado em distâncias adaptativas com vetores de pesos diferentes para os limites inferior e superior fica com as seguintes modificações:

1. Inicialização
   Fixe o número de agrupamentos K $(2 \leq K < n)$. Faça $\lambda_{jkL} = \lambda_{jkU} = 1$ $\forall j \in \{1, 2, \ldots, p\}$ e $\forall k \in \{1, 2, \ldots, K\}$. Escolha aleatoriamente uma partição inicial P de Ω em K grupos $P_1, \ldots, P_K$ ou, alternativamente, escolha aleatoriamente K observações $\boldsymbol{g}_1, \ldots, \boldsymbol{g}_K$ pertencendo a Ω como protótipos iniciais e aloque cada observação i de acordo com o protótipo mais próximo $\boldsymbol{g}_h$ $(h = \arg\min_{1 \leq k \leq K} \sum_{j=1}^{p} [\lambda_{jkL}(x_{iL_j} - \alpha_{kj})^2 + \lambda_{jkU}(x_{iU_j} - \beta_{kj})^2])$ para obter a partição inicial $P = \{P_1, \ldots, P_K\}$;

2. Atualização dos protótipos
   Serão atualizadas as componentes α_{kj} e β_{kj} nos protótipos dos grupos $\mathbf{g}_k$ $(k = 1, \ldots, K)$, conforme as Equações (2.38) e (2.39).

3. Atualização das distâncias

Os pesos serão calculados conforme as equações (2.43) e (2.44) e as distâncias atualizadas de acordo com a Equação (2.36).

4. Atualização da partição

   $test \leftarrow 0$

   para $i = 1$ até n faça

   defina o grupo vencedor P_h tal que

   $$h = \arg\min_{k=l,\ldots,K} = \sum_{j=1}^{p} [\lambda_{jkL}(x_{iL_j} - \alpha_{kj})^2 + \lambda_{jkU}(x_{iU_j} - \beta_{kj})^2]$$

   se $i \in P_k$ e $h \neq k$

   $test \leftarrow 1$

   $P_h \leftarrow P_h \cup \{i\}$

   $P_k \leftarrow P_k \setminus \{i\}$.

5. Critério de parada

   Se $test = 0$, então pare, caso contrário, volte ao passo (2)

Capítulo 3

Métodos Propostos

Estudos para encontrar a melhor forma de agrupar observações são abundantes, como Jain (2010), Xu e Tian (2015), Hammouda e Karray (2000), Kaufman e Rousseau (2005), Anderberg (1973), entre eles encontramos também uma grande variedade dos que dão foco ao agrupamento de séries temporais, tais como Box (1970), Kalpakis, Gada, e Puttagunta (2001), Aghabozorgi, Shirkhorshidi, e Wah (2015). Em menor número, encontramos as formas de agrupar séries temporais de dados tipo intervalo: Chavent e Lechevallier (2002),Carvalho, Brito, e Bock (2006). O algoritmo mais utilizado para esse tipo de agrupamento é o K-Means e, em Maharaj, Teles, e Brito (2019), encontramos várias medidas de distância para realizar esses agrupamentos.

As distâncias mais comuns utilizadas no agrupamento de dados são as distâncias euclidiana, Hausdorff, Mallows, dentre outras. Esse trabalho propõe apresentar algumas modificações na metodologia apresentada em Maharaj, Teles, e Brito (2019), as quais são baseadas em distâncias adaptativas e agrupamento baseado em kernel, com o objetivo de melhorar a performance de agrupamento de séries temporais.

Em aplicações práticas, é comum que diferentes variáveis influenciem um método de aprendizado de máquina de formas diferentes, por exemplo, uma variável x_1 pode ter uma maior importância na determinação de grupos de observações do que uma outra variável x_2. Distâncias adaptativas são recalculadas a cada iteração do algoritmo e permitem o aprendizado dos pesos das variáveis a partir dos dados. Isto é, um algoritmo de agrupamento baseado em uma distância adaptativa pode aprender a partir dos dados quais variáveis são importantes na definição dos grupos de observações e quais não são, calculando os pesos das variáveis de forma iterativa e como parte de um processo de otimização de uma função objetivo adequada.

Outra situação comum em problemas práticos é quando os grupos apresentam formas arbitrárias e/ou são não-linearmente separáveis. Nesse caso, algoritmos baseados na distância euclidiana e em suas adaptações podem apresentar um desempenho insatisfatório, visto que o uso da distância euclidiana pressupõe grupos aproximada-

mente linearmente separáveis e com formas aproximadamente hiper-esféricas. Em situações mais complexas, o uso de versões kernelizadas da distância euclidiana pode trazer ganhos. A ideia que sustenta os métodos baseados em kernel é a de que existe um mapeamento não-linear arbitrário do espaço original dos dados para um espaço de mais alta dimensão (possivelmente infinita), no qual grupos não-linearmente separáveis tornam-se separáveis. Na prática, distâncias euclidianas são calculadas no espaço original através de funções kernel e conseguimos definir hiper-superfícies de separação entre os grupos.

Levando em consideração o pequeno número de métodos de agrupamento para séries temporais com dados tipo intervalo, apresento neste capítulo novos algoritmos de agrupamentos de séries temporais tipo-intervalo baseados em distâncias adaptativas e a partir da kernelização da métrica e do espaço de características.

3.1 Novos Algoritmos de Agrupamento de Séries Temporais Intervalares

Maharaj, Teles, e Brito (2019) propõem o agrupamento de séries temporais intervalares a partir de algumas abordagens, a saber: (i) considera a informação de descritores estatísticos ou matriz de características (FM) extraídos a partir do centro e do raio (ou apenas do centro): (ii) a partir dos coeficientes estimados de modelos STAR e (iii) a partir dos valores da matriz de autocorrelação e funções de autocorrelação para o intervalo.

Na sequência desse estudo, escolhemos o método não-hierárquico K-Means aplicando conceitos de funções Kernel e distâncias adaptativas.

3.1.1 Agrupamento baseado em descritores estatísticos

Maharaj, Teles, e Brito (2019) utilizam os descritores estatísticos média, variância, assimetria, curtose e tendência dos centros e raios da série para montar uma matriz de características (Feature Matrix - FM) como citado na subseção 2.4.4. Utilizaremos aqui a mesma forma de extração de características, sendo o centro a soma dos limites dos intervalos dividida por dois; e o raio encontrado a partir da subtração do limite superior menos o limite inferior dividida por dois.

Utilizamos a linguagem R para encontrar os descritores e com isso montar a Feature Matrix (FM).

Como citamos anteriormente, o método de agrupamento que vamos utilizar será o K-Means, assim como Maharaj, Teles, e Brito (2019), porém, propomos diferentes distâncias.

Agrupamento através do algoritmo Kernel K-Means com kernelização da métrica

O algoritmo é bastante similar ao do K-Means, porém os protótipos iniciais são kernelizados, já que as distâncias entre as observações e os centroides serão calculadas a partir da kernelização da métrica. Esse algoritmo é citado nos resultados como Kernel K-Means.

1. Inicialmente, aplica-se o modelo de extração de características para montar a matriz de características, onde o número de linhas será a quantidade de séries e as colunas serão dos descritores estatísticos: média, variância, assimetria, curtose e tendência. Sendo as 5 primeiras colunas os descritores aplicados na informação do raio e as 5 colunas finais os descritores aplicados no centro da série.

2. Após fixar o número de clusters maior ou igual a dois e menor que o número total de intervalos, são escolhidos os protótipos (centroides) iniciais já no espaço kernelizado $(\Phi(\mathbf{g}_i), \ldots, \Phi(\mathbf{g}_k))$ e os conjuntos de intervalos serão alocados nesses centroides mais próximos através da equação $h = arg\ min_{1 \leq k \leq K} ||\Phi(\mathbf{x}_i) - \Phi(\mathbf{g}_k^{(t)})||^2$. Formando assim a partição inicial.

3. Conforme explicado na subseção 2.2.2, é feito a cada iteração o cálculo das novas distâncias, mas como se trata do K-Means Kernelizado (Kernel K-Means), as componentes α_{kj} e β_{kj} do centroide serão também kernelizadas conforme as equações (2.28) e (2.29).

4. Após isso, há uma atualização da partição de acordo com a regra de afetação dada na equação (2.30) onde os cálculos das distâncias já se encontram kernelizados, e isso se repete até que a função objetivo dada na equação (2.27) seja a mínima e não haja mais mudança de grupos dos intervalos.

Agrupamento através do algoritmo Kernel K-Means no espaço de características

O algoritmo baseado na kernelização do espaço de características foi brevemente explanado na subseção 2.6.2 é muito parecido com o da kernelização da métrica, mas aqui não será possível obter os protótipos da equação dos centroides porque o espaço kernelizado não é conhecido. Esse algoritmo é citado nos resultados como FS Kernel.

1. Na mesma matriz citada no item anterior, será aplicado este algoritmo.

2. Após fixar o número de clusters, a partição inicial P é escolhida de forma aleatória, já que não é possível obter os protótipos através da equação dos centroides, assim, o cálculo dos centroides se dará pela equação (2.32) e também não haverá atualização dos protótipos.

3. O cálculo das distâncias será através da equação (2.31).

4. Após isso, a atualização da partição de acordo com a regra de afetação dada na equação (2.34) onde são calculadas as distâncias dos intervalos kernelizados, e isso se repete até que a função objetivo dada na equação (2.33) seja a mínima e não haja mais mudança de grupos dos intervalos.

Agrupamento através do algoritmo K-Means com distâncias adaptativas

Aqui precisamos incluir uma nova etapa no algoritmo, já que serão dados pesos às observações de acordo com sua importância para o resultado. De acordo com esses pesos, as distâncias são adaptadas. Esse algoritmo é citado nos resultados como K-Means DA.

1. Na matriz de descritores estatísticos citada, aplico este algoritmo.

2. Após fixar o número de clusters são escolhidos os protótipos iniciais $(\boldsymbol{g}_i, \ldots, \boldsymbol{g}_k)$. A atualização dos protótipos é feita através das equações(2.38) e (2.39) para encontrar os componentes α_{kj} e β_{kj}.

3. É nessa etapa do algoritmo que há uma das mudanças em relação ao K-Means original, já que é preciso atualizar as distâncias conforme os pesos que são calculados na equação (2.40).

4. Após isso, há uma atualização da partição de acordo com a a equação (2.35) e esse é outro ponto de diferença em relação ao K-Means original, já que aqui não há regra de afetação.

5. Todo esse processo se repete até que a função objetivo dada na equação (2.37) seja a mínima e não haja mais mudança de grupos dos intervalos.

3.1.2 Agrupamento baseado nos coeficientes do modelo STAR

Um dos métodos de agrupamento de séries temporais intervalares utilizados por Maharaj, Teles, e Brito (2019) é baseado na extração dos coeficientes do modelo autorregressivo espaço-tempo, STAR na sigla em inglês.

Conforme apresentado na seção 2.4.4, o objetivo agora é apresentar as distâncias adaptativas, distâncias obtidas através da kernelização da métrica e distâncias obtidas a partir da kernelização do espaço de características. A equação utilizada para extrair os coeficientes é a equação (2.17). A extração dos coeficientes foi realizada através da linguagem R e com isso montamos a matriz de coeficientes.

Nessa matriz são aplicados os métodos de agrupamento K-Means original e os métodos aqui propostos, que são descritos mais detalhadamente abaixo.

Agrupamento através do algoritmo Kernel K-Means com kernelização da métrica

1. Inicialmente, aplica-se o modelo STAR e coletam-se os coeficientes. Com eles, é montada a nova matriz de informação para agrupamento, onde o número de linhas será a quantidade de séries e cada coluna será de um respectivo coeficiente.

2. Após isso, fixa-se o número de grupos maior ou igual a dois e menor que o número total de intervalos, são escolhidos os protótipos (centroides) iniciais já no espaço kernelizado $(\Phi(\mathbf{g}_i), \ldots, \Phi(\mathbf{g}_k))$ e os conjuntos de intervalos serão alocados nesses centroides mais próximo através da equação $h = arg\ min_{1\leq k\leq K}||\Phi(\mathbf{x}_i) - \Phi(\mathbf{g}_k^{(t)})||^2$. Formando assim a partição inicial.

3. Conforme explicado na subseção 2.2.2, é feito a cada iteração o cálculo das novas distâncias, mas como se trata do K-Means Kernelizado (Kernel K-Means), as componentes α_{kj} e β_{kj} do centroide serão também kernelizadas conforme as equações (2.28) e (2.29).

4. Então, há uma atualização da partição de acordo com a regra de afetação dada na equação (2.30) onde os cálculos das distâncias já se encontram kernelizados, e isso se repete até que a função objetivo dada na equação (2.27) seja a mínima e não haja mais mudança de grupos dos intervalos.

Agrupamento através do algoritmo Kernel K-Means no espaço de características

1. Na mesma matriz citada no item anterior, será aplicado este algoritmo.

2. Após fixar o número de clusters, a partição inicial P é escolhida de forma aleatória, já que não é possível obter os protótipos através da equação dos centroides, assim, o cálculo dos centroides se dará pela equação (2.32) e também não haverá atualização dos protótipos.

3. O cálculo das distâncias será através da equação (2.31).

4. Após isso, a atualização da partição de acordo com a regra de afetação dada na equação (2.34) onde são calculadas as distâncias dos intervalos kernelizados, e isso se repete até que a função objetivo dada na equação (2.33) seja a mínima e não haja mais mudança de grupos dos intervalos.

Agrupamento através do algoritmo K-Means com distâncias adaptativas

1. Na matriz de coeficientes do método STAR, aplico este algoritmo.

2. Aqui precisamos incluir uma nova etapa no algoritmo, já que serão dados pesos às observações de acordo com sua importância para o resultado. De acordo com esses pesos, as distâncias são adaptadas.

3. Após fixar o número de grupos são escolhidos os protótipos iniciais $((\mathbf{g}_i,\ldots,\mathbf{g}_k))$. A atualização dos protótipos é feita através das equações (2.38) e (2.39) para encontrar os componentes α_{kj} e β_{kj}.

4. É nessa etapa do algoritmo que há uma das mudanças em relação ao K-Means original, já que é preciso atualizar as distâncias conforme os pesos que são calculados na equação (2.40).

5. Após isso, há uma atualização da partição de acordo com a a equação (2.35) e esse é outro ponto de diferença em relação ao K-Means original, já que aqui não há regra de afetação.

6. Todo esse processo se repete até que a função objetivo dada na equação (2.37) seja a mínima e não haja mais mudança de grupos dos intervalos.

Capítulo 4

Resultados e Discussões

No presente capítulo, os métodos propostos serão comparados com algoritmos existentes na literatura, considerado dados simulados e uma aplicação a dados reais. A implementação dos algoritmos, simulação Monte Carlo e aplicação a dados reais foram realizadas em linguagem `R`.

4.1 Estudos de simulação

Nesta seção apresentamos os resultados obtidos através de simulações de Monte Carlo, considerando os cenários propostos em Maharaj, Teles, e Brito (2019).

Os experimentos Monte Carlo foram inspirados no artigo de Maharaj, Teles, e Brito (2019), que apresentam 4 cenários diferentes com séries temporais tipo-intervalo, estacionárias, geradas a partir de modelos STAR (subseção 2.4.4).

Em cada cenário, supomos duas estruturas de clusters presente nos dados. Na primeira, os dois clusters contém 10 séries temporais tipo-intervalo cada. Na segunda, consideramos o primeiro cluster com 5 séries temporais tipo-intervalo e segundo com 15 séries.

O número de réplicas Monte Carlo em cada uma das 8 configurações foi 100 e tamanho de amostra foi de 2.000 observações para cada série temporal tipo intervalo. As séries temporais tipo-intervalo foram geradas a partir de modelos STAR com diferentes configurações. Ademais, consideramos duas abordagens para extração das características das séries. Tais características serão utilizadas como variáveis para o computo da matriz de distâncias: (i) a partir da matriz de características (FM) e (ii) a partir dos coeficientes de modelos de espaço-tempo (STAR).

Serão comparados os algoritmos de agrupamento de séries temporais apresentados na seção 2.2 com método K-Means, utilizando como métrica o Índice de Rand Ajustado.

Abaixo, estão descritos os cenários com os respectivos parâmetros utilizados:

Cenário 1

Configura-se por clusters com diferentes níveis e sobreposição. (representado na Figura 4.1). As séries temporais tipo-intervalo foram geradas a partir de processos STAR com mesma ordem, a partir dos seguintes parâmetros:

- Cluster 1: $p = 2$ com $\theta_U = 2.75$; $\theta_L = 0.75$; $\phi_1 = 0.4$; $\phi_2 =$ - 0.3; $\psi_0 = 0.45$; $\psi_1 =$ -0.45; $\psi_2 = 0.3$; $\alpha^2_{aU} = 0.25$; $\alpha^2_{aL} = 0.49$; $\alpha_{LU} = 0.3$.
- Cluster 2: $p = 2$ com $\theta_U = 5$; $\theta_L = 4$; $\phi_1 = 0.3$; $\phi_2 = 0.2$; $\psi_0 =$ -0.35; $\psi_1 =$ -0.45; $\psi_2 = 0.35$; $\alpha^2_{aU} = 0.36$; $\alpha^2_{aL} = 0.64$; $\alpha_{LU} = 0.4$.

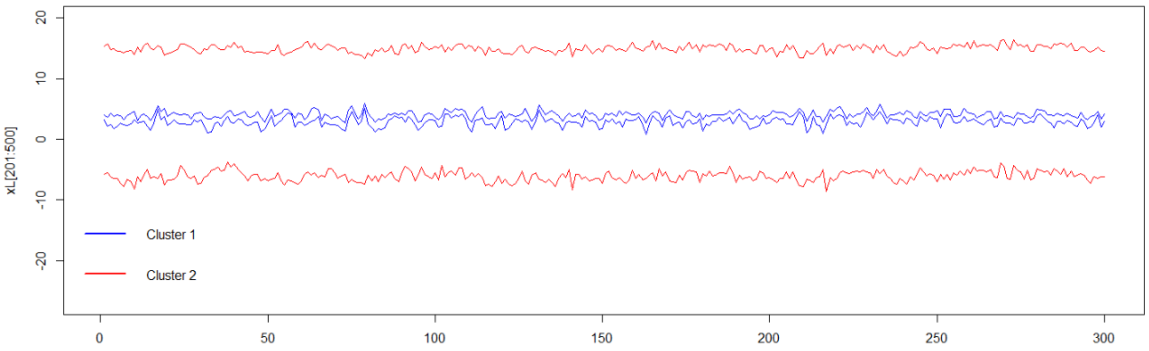

Figura 4.1: Apresentação de duas séries temporais tipo-intervalo representando a configuração do Cenário 1

Cenário 2

Configura-se por clusters com baixo grau de separação e dinâmicas diferentes. (representado na Figura 4.2). As séries temporais tipo-intervalo foram geradas a partir de processos STAR com diferentes ordens e níveis, a partir dos seguintes parâmetros:

- Cluster 1: $p = 1$ com $\theta_U = 2$; $\theta_L = 0.5$; $\phi_1 = 0.4$; $\psi_0 = 0.5$; $\psi_1 =$ -0.4; $\alpha^2_{aU} =$ 0.25; $\alpha^2_{aL} = 0.49$; $\alpha_{LU} = 0.25$.
- Cluster 2: $p = 2$ com $\theta_U = 6$; $\theta_L = 5$; $\phi_1 = 0.4$; $\phi_2 = 0.3$; $\psi_0 = 0.5$; $\psi_1 =$ -0.4; $\psi_2 =$ -0.3; $\alpha^2_{aU} = 0.64$; $\alpha^2_{aL} = 1$; $\alpha_{LU} = 0.36$.

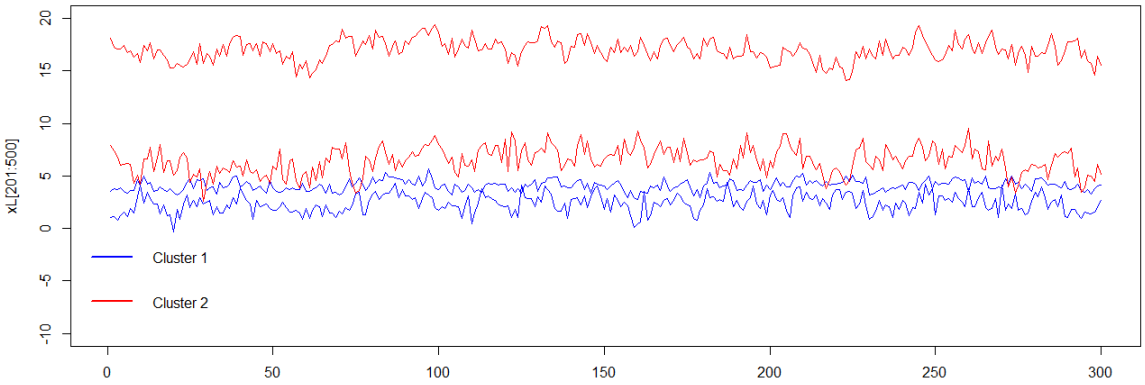

Figura 4.2: Apresentação de duas séries temporais tipo-intervalo representando a configuração do Cenário 2

Cenário 3

Configura-se por clusters com baixo grau de separação (representado na Figura 4.3), mas com alguma sobreposição e dinâmicas similares. As séries temporais tipo-intervalo foram geradas a partir de processos STAR com mesma ordem, parâmetros parecidos, diferentes níveis. Os parâmetros utilizados forma os seguintes:

- Cluster 1: $p = 2$ com $\theta_U = 4.25$; $\theta_L = 2.75$; $\phi_1 = 0.35$; $\phi_2 = 0.3$; $\psi_0 = 0.35$; ψ_1 = -0.35; $\psi_2 = 0.25$; $\alpha^2_{aU} = 0.25$; $\alpha^2_{aL} = 0.36$; $\alpha_{LU} = 0.15$.
- Cluster 2: $p = 2$ com $\theta_U = 2.5$; $\theta_L = 0.5$; $\phi_1 = 0.35$; $\phi_2 = 0.25$; $\psi_0 = 0.35$; ψ_1 = -0.3; $\psi_2 = 0.3$; $\alpha^2_{aU} = 0.25$; $\alpha^2_{aL} = 0.36$; $\alpha_{LU} = 0.15$.

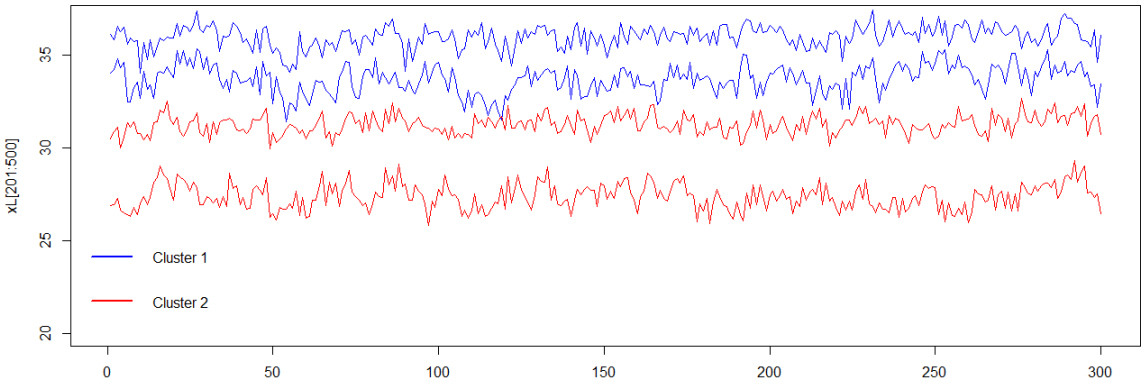

Figura 4.3: Apresentação de duas séries temporais tipo-intervalo representando a configuração do Cenário 3

Cenário 4

Configura-se por clusters com sobreposição (representado na Figura 4.4), e dinâmicas, parâmetros e níveis muito parecidos. As séries temporais tipo-intervalo foram geradas a partir de processos STAR com mesma ordem.

- Cluster 1: $p = 2$ com $\theta_U = 2.5$; $\theta_L = 0.75$; $\phi_1 = 0.35$;$\phi_2 = 0.3$; $\psi_0 = 0.35$; ψ_1 = -0.3; $\psi_2 = 0.25$; $\alpha^2_{aU} = 0.25$; $\alpha^2_{aL} = 0.36$; $\alpha_{LU} = 0.15$.
- Cluster 2: $p = 2$ com $\theta_U = 2.5$; $\theta_L = 0.5$; $\phi_1 = 0.35$;$\phi_2 = 0.25$; $\psi_0 = 0.35$; ψ_1 = -0.3; $\psi_2 = 0.3$; $\alpha^2_{aU} = 0.25$; $\alpha^2_{aL} = 0.36$; $\alpha_{LU} = 0.15$.

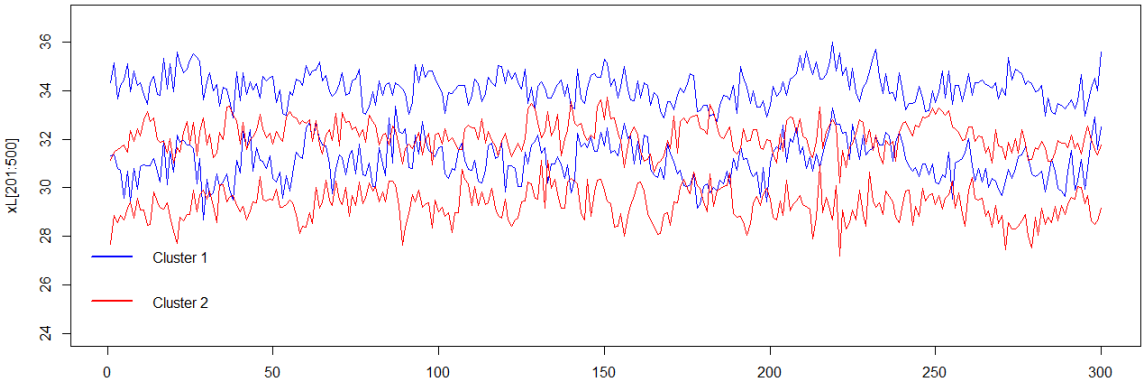

Figura 4.4: Apresentação de duas séries temporais tipo-intervalo representando a configuração do Cenário 4

4.1.1 Resultados dos experimentos

Nas tabelas a seguir, estão as médias e desvios padrão para o Índice de Rand Ajustado correspondente ao agrupamento das séries temporais tipo-intervalo, segundo a extração das características (FM ou modelo STAR) e o algoritmo de agrupamento. Foram considerados o centro e o raio das séries. Após obtidos os coeficientes do modelo STAR, aplicamos também uma padronização para avaliar se o efeito da escala dos coeficientes influenciam os resultados do agrupamento. Chamamos essa abordagem de *STAR scale*.

A configuração (C1:10, C2:10) se refere aos clusters 1 e 2 com 10 séries temporais tipo-intervalo cada. Já a configuração (C1:5, C2:15) se refere ao cluster 1 com 5 séries e ao cluster 2 com 15 séries.

Os novos algoritmos de agrupamento K-Means com Distâncias Adaptativas (K-Means DA), K-Means Kernelizado (Kernel K-Means) e K-Means com Kernelização do Espaço de Características (FS Kernel) foram comparados com o método K-Means (tradicional). Vale ressaltar que para cada um dos métodos utilizados, foram consideradas 100 partições iniciais e escolhida aquela que minimizou a função objetivo para cada método.

Tabela 4.1: Cenário 1: média e desvio-padrão (DP) para o índice de Rand ajustado segundo a abordagem, configurações dos clusters e o método de agrupamento.

Configuração C1:10 C2:10

Abordagem	Método				
	Estatística	K-means	K-Means DA	Kernel K-means	FS Kernel
FM	Média	1,000	1,000	1,000	1,000
	DP	0,000	0,000	0,000	0,000
STAR	Média	0,920	**0,995**	0,943	0,959
	DP	0,102	0,053	0,101	**0,096**
STAR scale	Média	0,837	**0,995**	0,838	0,845
	DP	0,079	0,053	**0,080**	0,064

Configuração C1:5 C2:15

Abordagem	Método				
	Estatística	K-means	K-Means DA	Kernel K-means	FS Kernel
FM	Média	1,000	1,000	1,000	0,999
	DP	0,000	0,000	0,000	0,010
STAR	Média	0,751	**0,943**	0,751	0,744
	DP	0,118	**0,155**	0,118	0,115
STAR scale	Média	0,851	**0,943**	0,860	0,804
	DP	0,146	**0,155**	0,149	0,140

Cenário 1

Observando as tabelas, conseguimos identificar resultados idênticos ao já conhecido algoritmo K-Means na configuração (C1:10, C2:10) e muito próximos na configuração (C1:5, C2:15) quando o método de extração de características é o de Matriz de Características *(Feature Matrix - FM)*. O que era esperado já que as séries são estacionárias. Com isso não há uma melhora considerável quando se trata de clusters bem separados de séries geradas temporais de dados tipo-intervalo a partir de processos com mesma ordem e diferentes níveis e dinâmicas.

Já quando se trata dos parâmetros STAR, na configuração (C1:10, C2:10), todos os novos métodos propostos apresentaram resultados melhores que o do K-Means, tendo um destaque para o K-Means com distâncias adaptativas *(K-Means DA)*. Já na configuração (C1:5, C2:15), os resultados foram muito parecidos, com exceção do K-Means com distâncias adaptativas que trouxeram ótimos resultados, quando os clusters são desbalanceados. Além disso, o STAR escalonado se saiu melhor nessa configuração.

Tabela 4.2: Cenário 2: média e desvio-padrão (DP) para o índice de Rand ajustado segundo a abordagem, configurações dos clusters e o método de agrupamento.

Configuração C1:10 C2:10

Abordagem	Método				
	Estatística	K-means	K-Means DA	Kernel K-means	FS Kernel
FM	Média	1,000	1,000	1,000	1,000
	DP	0,000	0,000	0,000	0,000
STAR	Média	0,879	0,945	0,875	**0,964**
	DP	0,109	0,085	0,096	**0,088**
STAR scale	Média	0,671	**0,945**	0,690	0,715
	DP	0,127	**0,085**	0,119	0,074

Configuração C1:5 C2:15

Abordagem	Método				
	Estatística	K-means	K-Means DA	Kernel K-means	FS Kernel
FM	Média	1,000	1,000	1,000	1,000
	DP	0,000	0,000	0,000	0,000
STAR	Média	0,547	**0,859**	0,547	0,565
	DP	0,087	**0,145**	0,080	0,066
STAR scale	Média	0,717	**0,859**	0,755	0,627
	DP	0,178	0,145	0,203	0,115

Cenário 2

Quando utilizamos a Matriz de Características, não há nenhuma melhora considerável quando se trata de clusters bem separados de séries temporais tipo-intervalo

geradas a partir de processos com diferentes ordem, níveis e dinâmicas, os resultados se mostraram idênticos ao já conhecido algoritmo K-Means tanto na configuração de 10 séries em cada cluster *(C1:10 C2:10)* quanto na configuração de 5 séries em um cluster e 15 séries em outro cluster *(C1:5 C2:15)*.

Porém quando se trata da forma de extração utilizando os parâmetros STAR, em ambas as configurações, independentemente de estar escalonado ou não, o K-Means com distâncias adaptativas teve excelentes resultados, acima dos demais métodos.

Tabela 4.3: Cenário 3: média e desvio-padrão (DP) para o índice de Rand ajustado segundo a abordagem, configurações dos clusters e o método de agrupamento.

Configuração C1:10 C2:10

Abordagem	Método				
	Estatística	K-means	K-Means DA	Kernel K-means	FS Kernel
FM	Média	0,993	**1,000**	0,994	0,986
	DP	0,035	0,000	0,027	0,039
STAR	Média	0,500	0,501	0,500	**0,502**
	DP	0,034	0,036	0,034	**0,037**
STAR scale	Média	0,499	0,501	0,499	**0,501**
	DP	0,036	0,036	0,036	**0,037**

Configuração C1:5 C2:15

Abordagem	Método				
	Estatística	K-means	K-Means DA	Kernel K-means	FS Kernel
FM	Média	0,998	**1,000**	0,998	0,931
	DP	0,014	0,000	0,014	0,092
STAR	Média	**0,518**	0,516	**0,518**	0,513
	DP	0,049	0,047	**0,055**	0,043
STAR scale	Média	0,520	0,516	0,518	0,518
	DP	0,055	0,047	0,051	**0,056**

Cenário 3

Quando se trata de clusters não tão bem separados de séries geradas a partir de processos com diferentes níveis, mesma ordem, dinâmicas um pouco diferentes e parâmetros parecidos, extraindo as características a partir da Feature Matrix, houve resultados melhores que os do K-Means em ambas as configurações no método utilizando as distâncias adaptativas (*K-Means DA*) e no método com K-Means Kernelizado (*Kernel K-Means*). O que não ocorreu com a Kernelização do Espaço de Características (*FS Kernel*).

Já extraindo as características a partir dos parâmetros STAR, em ambas as configurações, não houve melhora considerável e na configuração *(C1:5 C2:15)* o resultado foi até pior que no K-Means tradicional.

Tabela 4.4: Cenário 4: média e desvio-padrão (DP) para o índice de Rand ajustado segundo a abordagem, configurações dos clusters e o método de agrupamento.

Configuração C1:10 C2:10

Abordagem	Método				
	Estatística	K-means	K-Means DA	Kernel K-means	FS Kernel
FM	Média	0,997	**1,000**	0,996	0,990
	DP	0,017	0,000	0,020	**0,030**
STAR	Média	0,503	0,502	0,503	**0,506**
	DP	0,035	0,036	0,035	0,034
STAR scale	Média	0,500	0,502	0,501	**0,502**
	DP	0,038	0,036	0,037	**0,040**

Configuração C1:5 C2:15

Abordagem	Método				
	Estatística	K-means	K-Means DA	Kernel K-means	FS Kernel
FM	Média	0,999	**1,000**	0,999	0,939
	DP	0,010	0,000	0,010	**0,113**
STAR	Média	0,508	**0,511**	0,507	0,504
	DP	0,046	**0,061**	0,047	0,051
STAR scale	Média	0,512	0,511	0,510	0,509
	DP	0,057	**0,061**	0,055	**0,061**

Cenário 4

Quando se trata de clusters mal separados de séries geradas a partir de processos com mesma ordem e níveis, e dinâmicas muito parecidas, a forma de extração FM resultou em números melhores que o K-Means em ambas as configurações apenas com as distâncias adaptativas. Os demais métodos apresentados não trouxeram vantagem.

Os métodos aplicados na matriz gerada a partir dos parâmetros STAR, na configuração *(C1:10 C2:10)* trouxeram melhores resultados com um destaque para o FS Kernel. Já na configuração *(C1:5 C2:15)*, o melhor resultado foi com Distâncias adaptativas sem escalonamento em ambas as configurações. Com escalonamento, apenas a configuração *(C1:10 C2:10)* apresentou melhores resultados.

Acreditamos que a melhor performance dos métodos para a forma de extração FM se dá pelo fato de que as séries simuladas serem estacionárias. Como trabalho futuro, pretendemos realizar Um estudo de simulação envolvendo com séries não-estacionárias e com presença de sazonalidade.

4.2 Aplicação a dados reais

Após avaliar a performance dos métodos em dados simulados, nesta seção apresentamos uma aplicação dos métodos presentes nesta dissertação a bases de dados reais. As séries temporais tipo-intervalo analisadas se dividem em: índices financeiros e criptomoedas. Ademais, trazemos as séries com frequência diária, semanal e mensal, para um intervalo de um ano.

Serão observados 11 índices financeiros, a saber: IBOVESPA, Petróleo (Crude Oil), Dow Jones, Euro, NASDAQ, FTSE, Nikkei, Prata, Ouro, Russel e SP500; Em relação às 35 criptomoedas, foram consideradas aquelas com os maiores valores de mercado na data da extração dos dados (23/10/2021), a saber: Algorand(ALGO), Arweave(AR), Avalanche(AVAX), BinanceCoin(BNB), Bitcoin(BTC), BitcoinCash(BCH), BitcoinSV(BSV), Cardano(ADA), Chainlink(LINK), Cosmos(ATOM1), Creditcoin(CTC1), Dash(DASH), Dogecoin(DOGE), Elrond(EGLD), EOS(EOS), Ethereum(ETH), EthereumClassic(ETC), HederaHashgraph(HBAR), Helium(HNT1), IOTA(MIOTA), Kusama(KSM), Litecoin(LTC), Monero(XMR), NEO(NEO), Polkadot(DOT1), Solana(SOL1), Stellar(XLM), Terra(LUNA1), Tether(USDT), Tezos(XTZ), THETA(THETA), TRON(TRX), VeChain(VET), Waves(WAVES) e XRP(XRP).

As criptomoedas surgiram há relativamente pouco tempo e já se tornaram parte da economia mundial, sendo a mais famosa dela o Bitcoin, que é inclusive aceito para realizar compras em muitos estabelecimentos, principalmente virtuais. Hoje, existem mais de 500 criptomoedas no mundo, sendo a maioria delas ligada a algum projeto. Apesar do descrédito por parte de muitos, principalmente por se tratarem de moedas "virtuais", é interessante citar que o dinheiro como conhecemos já é em grande parte virtual. Isso porque apenas uma pequena parte do dinheiro que existe em todo mundo é manipulado fisicamente. A principal diferença das cripto-moedas está aí, não existe versão física delas.

É importante citar que as cripto-moedas não são emitidas por nenhum país como o Real, o Dólar e o Euro. Entretanto, entender seu relacionamento com índices financeiros pode representar uma informação valiosa. É interessante pensar nas criptomoedas como um dinheiro de circulação livre, sem que você precise ficar vendendo ou comprando, realizando câmbios, como nas moedas tradicionais.

Em primeiro lugar, iremos considerar os índices financeiros citados acima, a partir de médias mensais pelo período de 15 anos, entre Novembro de 2006 e Outubro de 2021. Em seguida, objetivamos agrupar os índices financeiros e as principais criptomoedas, observadas no período de 1 ano (de 26/10/2020 a 22/10/2021), considerando uma frequência diária, semanal e mensal (apenas dias úteis foram consi-

derados, visto que criptomoedas que possuem cotações todos os dias). Dessa forma, quatro bases de dados foram analisadas.

Além dos valores brutos das séries, também foi aplicado o log retorno, um índice utilizado para fazer comparações mais diretas entre números com ordem de grandezas diferentes. Entretanto, chegamos à conclusão que os resultados obtidos não apresentaram diferenças significativas. Dessa forma, os resultados apresentados nesta seção consideram os valores observados das variáveis em questão.

4.2.1 Resultados da aplicação

Serão considerados os seguintes índices internos para comparação dos algoritmos de agrupamento: Silhueta, Dunn e Conectividade, que foram descritos na subseção 2.3.4. Como não conhecemos a partição *a priori*, o número de clusters (K) foi determinado através do método do cotovelo. É importante citar que K não será, necessariamente, igual em todos os métodos. Dessa forma, para cada método de agrupamento foi aplicado o método do cotovelo e identificado o valor ótimo de K.

Os resultados serão apresentados em Tabelas, levando em conta cada índice interno, cada algoritmo de agrupamento (K-means e os 4 métodos propostos) e a abordagem de extração das variáveis (FM ou coeficientes do modelo STAR). Uma vez escolhido o método com o melhor agrupamento, iremos apresentar os grupos com os respectivos índices financeiros (em azul) e criptomoedas (em vermelho), bem como sua representação gráfica.

Antes de entrarmos no detalhe de cada série, concluíamos que a melhor abordagem de extração de característica é o Feature Matrix (FM), que apresentou resultados superiores para todas as séries, algoritmos e índices internos, se comparado ao utilização dos coeficientes do modelo STAR. Tal resultado está em concordância com os obtidos nos dados simulados.

Dessa forma, iremos dar maior ênfase nesta abordagem na comparação entre os métodos, em nossos resultados. É interessante também citar que os melhores métodos estão destacados nas tabelas.

Indices Financeiros

Os resultados estão dispostos nas tabelas separados pelos índices: Conectividade, Dunn e Silhueta. Nas séries mensais de 15 anos, de acordo com a tabela 4.5 observamos que o índice Conectividade (quanto menor melhor o método) nos traz como melhor algoritmo o K-means na abordagem de extração de dados Feature Matrix. Esta será a única abordagem citadas nestes resultados, visto que apresentou resultados superiores as duas outras abordagens em todos os casos. De acordo com o Índice Dunn (quanto mais próximo de 1 melhor o método), tmeos que o melhor

algoritmo foi o Kernel K-Means. Já avaliando a Silhueta (quanto mais próximos de 1 melhor o método), temos como melhor método o FS Kernel.

Tabela 4.5: Dados mensais de 15 anos: segundo a abordagem, configurações dos clusters e o método de agrupamento.

Índice Conectividade

Abordagem	Método			
	K-means	K-Means DA	Kernel K-means	FS Kernel
FM	18,334	25,902	23,385	23,792
STAR	24,572	29,431	29,016	28,033
STAR scale	27,790	29,431	28,633	27,025

Índice Dunn

Abordagem	Método			
	K-means	K-Means DA	Kernel K-means	FS Kernel
FM	0,528	0,543	0,636	0,552
STAR	0,200	0,257	0,200	0,292
STAR scale	0,200	0,257	0,200	0,292

Índice Silhueta

Abordagem	Método			
	K-means	K-Means DA	Kernel K-means	FS Kernel
FM	0,132	0,149	0,165	0,289
STAR	-0,016	-0,291	-0,291	-0,291
STAR scale	-0,291	-0,291	-0,291	-0,291

Como cada um dos índices trouxe um método como melhor, foi um utilizado um ranqueamento. Como podemos ver na tabela 4.6, o método que trouxe o melhor resultado de acordo com os índices foi o Kernel K-means. Isso faz sentido, visto que as séries apresentam algumas observações destoantes das demais (*outliers*). Dessa forma, o método K-means com distância euclidiana pode não ser eficiente se comparada uma distância kernelizada. Abaixo, vemos como ficaram agrupados os índices financeiros para algoritmo K-means com distância kernelizada. Lembrando que não consideramos as criptomoedas neste cenário de média de 15 anos.

Tabela 4.6: Ranking dos métodos nos índices de comparação das séries de índices financeiros de 15 anos.

Índice	K-Means	K-Means DA	Kernel K-Means	FS Kernel
Conectividade	1	4	2	3
Dunn	4	3	1	2
Silhueta	4	3	2	1
Soma	**9**	**10**	**5**	**6**

A figura 4.5 ilustra a distribuição das séries temporais tipo-intervalo para os índices financeiros nos grupos obtidos de acordo com o algoritmo Kernel K-means a partir da matriz de descritores estatísticos. Os eixos horizontal e vertical correspondem, respectivamente, à primeira e segunda componentes principais extraídas da matriz de descritores estatísticos. Entre parênteses nos rótulos dos eixos estão apresentadas as porcentagens da variância explicadas por cada componente. O método do cotovelo propôs 6 grupos e assim podemos notar grupos bem separados e sem sobreposição.

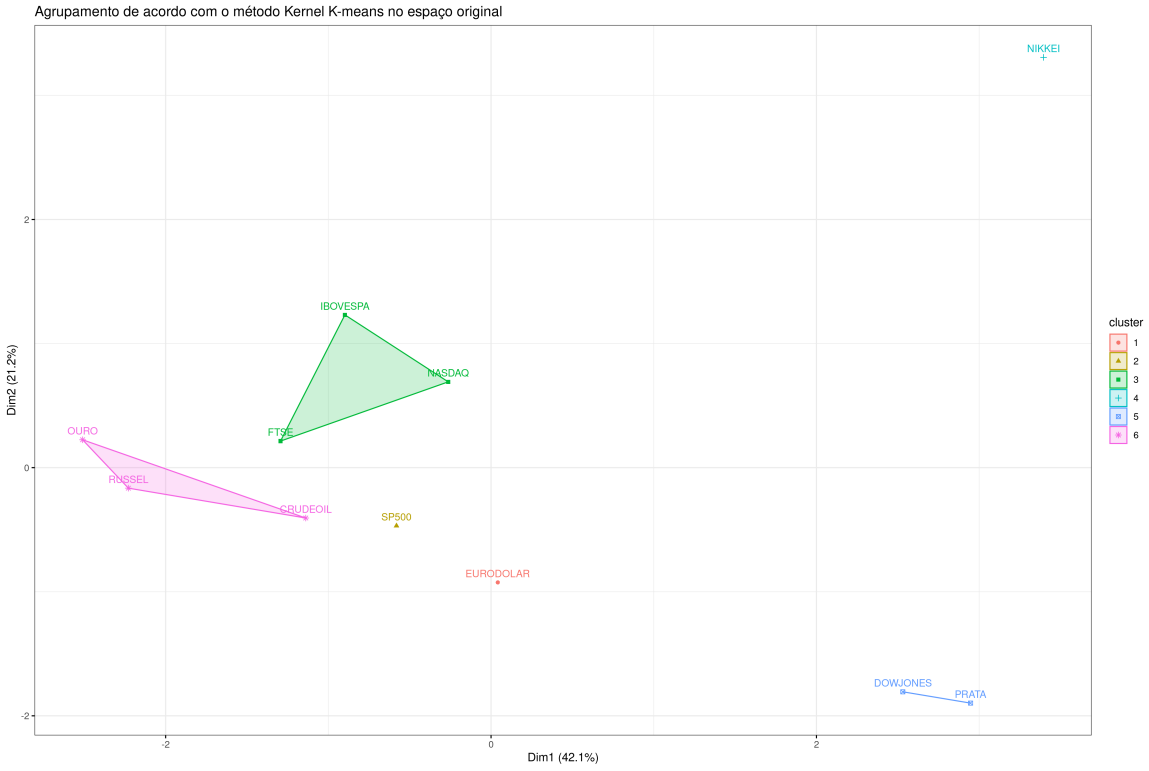

Figura 4.5: Representação gráfica do agrupamento dos índices financeiros obtidos pelo método Kernel K-means - Dados mensais de 15 anos

A tabela 4.7 apresenta a lista dos índices financeiros em cada cluster. Visto que

o Euro é a única moeda, faz sentido que esteja em um grupo a parte. O SP500 também porque ele reúne ações das 500 maiores empresas do mundo. Dessa forma, acaba sendo uma mistura de tudo e não se relaciona diretamente com nenhum dos demais índices. No grupo 3, vemos as bolsas de valores de 3 das maiores cidades do mundo: IBOVESPA de São Paulo, NASDAQ de Nova Iorque e FTSE100 de Londres. Já o NIKKEI, também ficou em um cluster isolado. O índice Dow Jones aparece agrupado junto com a Prata. Por último, o petróleo (Crude Oil) aparece junto com ouro e com o Russel, que reúne ações de 2000 empresas de pequena capitalização nos Estados Unidos.

Tabela 4.7: Agrupamentos dos índices financeiros obtidos pelo método Kernel K-means - Dados mensais de 15 anos.

Índices	**Grupos**
Euro	1
SP500	2
IBOVESPA	3
NASDAQ	3
FTSE	3
NIKKEI	4
DOW JONES	5
PRATA	5
CRUDE OIL	6
OURO	6
RUSSEL	6

Dados diários

Nas séries temporais tipo intervalo com cotações diárias, analisamos a possível relação entre os índices financeiros e as criptomoedas. Para este conjunto de dados, segundo a tabela 4.8, o índice Conectividade avaliou como melhor método o K-Means DA, já o Índice Dunn trouxe o K-Means como sendo o melhor, e na Silhueta foi o Kernel K-Means.

Tabela 4.8: Dados diários de um ano: segundo a abordagem, configurações dos clusters e o método de agrupamento.

Índice Conectividade

Abordagem	Método			
	K-means	K-Means DA	Kernel K-means	FS Kernel
FM	46,825	15,406	46,737	32,490
STAR	116,151	86,559	106,140	99,289
STAR scale	115,576	86,559	112,977	90,502

Índice Dunn

Abordagem	Método			
	K-means	K-Means DA	Kernel K-means	FS Kernel
FM	0,216	0,105	0,162	0,149
STAR	0,013	0,015	0,015	0,015
STAR scale	0,013	0,015	0,013	0,012

Índice Silhueta

Abordagem	Método			
	K-means	K-Means DA	Kernel K-means	FS Kernel
FM	0,167	0,094	0,311	0,050
STAR	-0,017	0,143	-0,204	-0,283
STAR scale	0,142	0,143	0,104	-0,35

Na tabela 4.9, trazemos o ranking dos métodos levando em consideração a abordagem Feature Matrix. Na última linha, o somatório foi realizado para escolher o método com menor ranking. O algoritmo que apresentou o melhor resultado, de acordo com o ranking obtido a partir dos índices internos, foi o Kernel K-means.

Tabela 4.9: Ranking dos métodos nos índices de comparação das séries diárias.

Índice	K-Means	K-Means DA	Kernel K-Means	FS Kernel
Conectividade	4	1	3	2
Dunn	1	4	2	3
Silhueta	2	3	1	4
Soma	**7**	**8**	**6**	**9**

A figura 4.6 ilustra como as séries diárias foram agrupadas utilizando o método Kernel K-means. O método do cotovelo propôs 10 grupos. Podemos notar 7 grupos bem separados e 3 com sobreposição.

Figura 4.6: Representação gráfica do agrupamento dos índices e criptomoedas obtidos pelo método Kernel K-means - Dados diários de um ano

A tabela 4.10, apresenta a relação dos índices financeiros e criptomoedas em cada cluster. Os clusters apresentaram quantidades diferentes de séries. Ademais, observou-se clusters com presença de apenas criptomoedas (1, 5, 6, 7, 8 e 9), apenas índices financeiros (4 e 10) e com a presença de ambas. O Bitcoin e o Ethereum, apesar de serem as criptomoedas com maior mercado em capital, não aparecem no mesmo cluster.

Tabela 4.10: Agrupamentos dos índices financeiros e criptomoedas obtidos pelo método Kernel K-means - Dados diários de um ano.

Série	Grupos	Série	Grupos
Dash	1	S&P500	3
Dogecoin	1	FTSE	4
BinanceCoin	2	LIBRA	4
BitcoinSV	2	Tether	5
Monero	2	VeChain	5
XRP	2	EOS	6
Tezos	2	EthereumClassic	6
CRUDEOIL	2	Polkadot	7
DOWJONES	2	Elrond	7
EURO	2	Ethereum	7
OURO	2	HederaHashgraph	7
PRATA	2	NEO	8
Cardano	3	Solana	8
Algorand	3	Avalanche	9
Arweave	3	BitcoinCash	9
Cosmos	3	Helium	9
Bitcoin	3	Kusama	9
Creditcoin	3	Chainlink	9
Terra	3	Litecoin	9
IOTA	3	Waves	9
THETA	3	Stellar	9
TRON	3	NASDAQ	10
RUSSEL	3	NIKKEI	10

Dados semanais

Nas séries temporais tipo intervalo com cotações semanais, também analisamos a possível relação entre os índices financeiros e as criptomoedas. De acordo com os índices Conectividade e Silhueta, o método que apresentou melhor resultado para o conjunto de dados com frequência semanal foi o K-Means, e segundo o Índice Dunn foi o Kernel K-Means (Tabela 4.11).

Tabela 4.11: Dados semanais de um ano: segundo a abordagem, configurações dos clusters e o método de agrupamento.

Índice Conectividade

Abordagem	Método			
	K-means	K-Means DA	Kernel K-means	FS Kernel
FM	38,958	43,493	48,378	44,314
STAR	112,548	105,387	117,248	111,648
STAR scale	111,858	105,387	112,891	108,852

Índice Dunn

Abordagem	Método			
	K-means	K-Means DA	Kernel K-means	FS Kernel
FM	0,194	0,174	0,253	0,085
STAR	0,030	0,029	0,030	0,030
STAR scale	0,030	0,029	0,031	0,030

Índice Silhueta

Abordagem	Método			
	K-means	K-Means DA	Kernel K-means	FS Kernel
FM	0,270	0,167	0,149	0,146
STAR	0,015	-0,294	-0,217	-0,049
STAR scale	-0,135	-0,294	-0,203	-0,269

Na tabela 4.12 trazemos o ranking dos métodos levando em consideração a abordagem Feature Matrix. Na última linha, o somatório foi realizado para escolher o método com menor ranking. O algoritmo que apresentou o melhor resultado, de acordo com o ranking obtido a partir dos índices internos, foi o K-means Por se tratarem das médias semanais das séries, os números não apresentam grandes mudanças entre si no período de um ano, assim, tal método utilizando a distância euclidiana atende bem a necessidade do agrupamento.

Tabela 4.12: Ranking dos métodos nos índices de comparação das séries semanais.

Índice	K-Means	K-Means DA	Kernel K-Means	FS Kernel
Conectividade	1	2	4	3
Dunn	2	3	1	4
Silhueta	1	2	3	4
Soma	**4**	**7**	**8**	**11**

O gráfico 4.7 ilustra como as séries semanais foram agrupadas quando utilizamos o método K-means. O método do cotovelo propôs 7 grupos e assim podemos notar 7 grupos não tão bem separados e com alguma sobreposição.

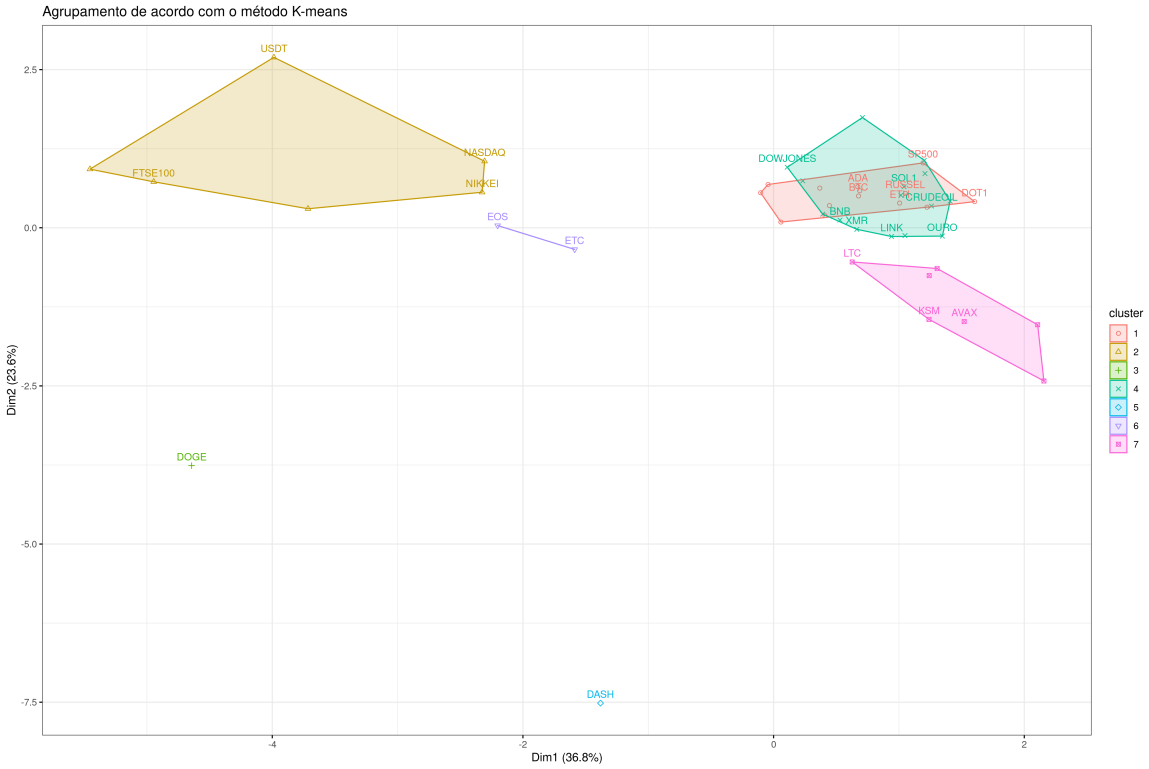

Figura 4.7: Representação gráfica do agrupamento dos índices e criptomoedas obtidos pelo método Kernel K-means - Dados semanais de um ano

Na tabela 4.13, não vemos grupos apenas com índices financeiros, mas vemos grupos com apenas um elemento como é o caso dos grupos 3 e 5, com as criptos Dogecoin e Dash que no agrupamento de dados diários (tabela 4.8) aparecem juntas sozinhas em um grupo. Há também muitos grupos mistos e os 6 e 7 formados apenas de criptos. Dessa vez o Bitcoin e o Ethereum aparecem juntos e o Bitcoin continua no mesmo grupo dos índices financeiros Russel e SP500. As criptos Cardano, Algorand, Arweave, Cosmos, Creditcoin e IOTA também apareceram nesse grupo nos índices

diários e semanais. As criptos Tether e VeChain que apareceram juntas sozinhas em um grupo nos índices diários, agoras constam como únicas criptos em um grupo com vários índices financeiros. O Petróleo (Crude Oil), Dow Jones, Euro, Ouro e Prata aparecem no mesmo grupo nos dados diários e agora nos semanais, mas não apareceram todos juntos nos dados mensais de 15 anos, mas sim em pares(tabela 4.5). As criptos Avalanche, BitcoinCash, Helium, Kusama, Litcoin, Waves e Stellar aparecem mais uma vez juntas.

Tabela 4.13: Agrupamentos dos índices financeiros e criptomoedas obtidos pelo método K-means - Dados semanais de um ano.

Série	Grupos	Série	Grupos
ADA	1	LINK	4
ALGO	1	NEO	4
AR	1	SOL1	4
ATOM1	1	THETA	4
BTC	1	TRX	4
CTC1	1	XMR	4
DOT1	1	XRP	4
EGLD	1	XTZ	4
ETH	1	CRUDEOIL	4
HBAR	1	DOWJONES	4
LUNA1	1	EURO	4
MIOTA	1	OURO	4
RUSSEL	1	PRATA	4
SP500	1	DASH	5
USDT	2	EOS	6
VET	2	ETC	6
FTSE	2	AVAX	7
LIBRA	2	BCH	7
NASDAQ	2	HNT1	7
NIKKEI	2	KSM	7
DOGE	3	LTC	7
BNB	4	WAVES	7
BSV	4	XLM	7

Dados mensais

Nas séries temporais tipo intervalo com cotações mensais, segundo a tabela 4.14, os índices Conectividade e Silhueta apresentaram os melhores resultados no método K-Means, já o índice Dunn considerou o Kernel K-Means como o melhor método de agrupamento.

Tabela 4.14: Dados mensais de um ano: segundo a abordagem, configurações dos clusters e o método de agrupamento.

Índice Conectividade

Abordagem	Método			
	K-means	K-Means DA	Kernel K-means	FS Kernel
FM	47,036	90,790	52,228	48,134
STAR	111,377	114,048	108,398	103,196
STAR scale	109,125	116,870	111,583	101,484

Índice Dunn

Abordagem	Método			
	K-means	K-Means DA	Kernel K-means	FS Kernel
FM	0,186	0,091	0,252	0,102
STAR	0,061	0,064	0,061	0,057
STAR scale	0,063	0,063	0,055	0,064

Índice Silhueta

Abordagem	Método			
	K-means	K-Means DA	Kernel K-means	FS Kernel
FM	0,272	-0,166	0,140	0,163
STAR	-0,157	-0,305	-0,034	-0,132
STAR scale	-0,091	-0,031	-0,079	-0,058

Na tabela 4.15, trazemos o ranking dos métodos levando em consideração a abordagem Feature Matrix. Na última linha, o somatório foi realizado para escolher o método com menor ranking. O melhor método de agrupamento foi o K-means, de acordo com o ranking obtido a partir dos índices internos.

Tabela 4.15: Ranking dos métodos nos índices de comparação das séries mensais.

Índice	K-Means	K-Means DA	Kernel K-Means	FS Kernel
Conectividade	1	4	3	2
Dunn	2	4	1	3
Silhueta	1	4	3	2
Soma	**4**	**12**	**7**	**7**

O gráfico 4.8 ilustra como as séries mensais dos foram agrupadas quando utilizamos o método K-means. O método do cotovelo propôs 8 grupos e assim podemos notar pelo menos 3 grupos mal separados e com sobreposição.

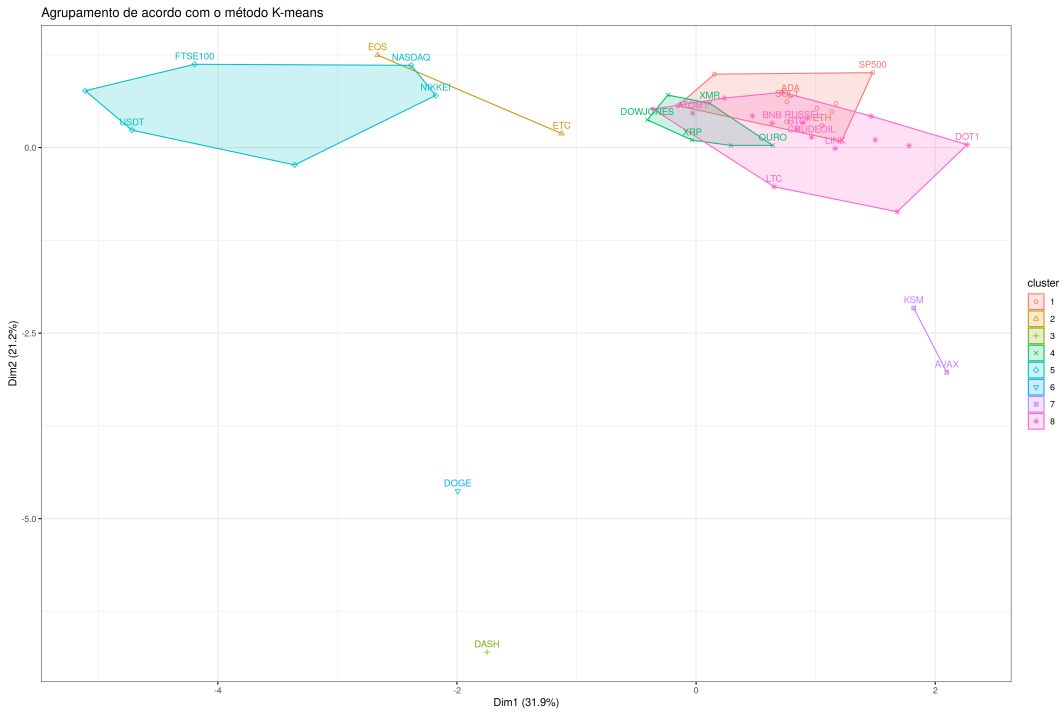

Figura 4.8: Representação gráfica do agrupamento dos índices e criptomoedas obtidos pelo método Kernel K-means - Dados mensais de um ano

Na tabela 4.16, mais uma vez vemos a cripto Dogecoin e a Dash sozinhas nos grupos 3 e 6 respectivamente. Não vemos grupos apenas com índices financeiros. Há também muitos grupos mistos e o 7 formado apenas de criptos. O Bitcoin e o Ethereum aparecem separados como nos dados de frequência diária (tabela 4.8) e o Bitcoin continua no mesmo grupo do índice financeiro Russel, mas o SP500 dessa vez está junto com a Ethereum. As criptos Cardano, Algorand, Arweave e Creditcoin também apareceram nesse grupo do Ethereum e SP500, já as Cosmos e IOTA continuaram acompanhando o Bitcoin e o Russel 2000. As criptos Tether e VeChain continuam como únicas criptos em um grupo com vários índices financeiros. O Dow Jones, Euro, Ouro e Prata aparecem no mesmo grupo nos dados diários,

semanais(tabela 4.14) e agora nos mensais, saindo desse grupo apenas o Petróleo (Crude Oil). As criptos BitcoinCash, Litecoin, Waves e Stellar aparecem mais uma vez juntas como nos dados de frequência diária e semanal.

Tabela 4.16: Agrupamentos dos índices financeiros e criptomoedas obtidos pelo método K-means - Dados mensais de um ano.

Série	Grupos	Série	Grupos
ADA	1	LIBRA	5
ALGO	1	NASDAQ	5
AR	1	NIKKEI	5
CTC1	1	DOGE	6
EGLD	1	AVAX	7
ETH	1	KSM	7
HBAR	1	ATOM1	8
HNT1	1	BCH	8
SOL1	1	BNB	8
TRX	1	BSV	8
SP500	1	BTC	8
EOS	2	DOT1	8
ETC	2	LINK	8
DASH	3	LTC	8
XMR	4	LUNA1	8
XRP	4	MIOTA	8
DOWJONES	4	NEO	8
EURO	4	THETA	8
OURO	4	WAVES	8
PRATA	4	XLM	8
USDT	5	XTZ	8
VET	5	CRUDEOIL	8
FTSE	5	RUSSEL	8

Finalmente, observamos que os novos métodos apresentaram uma melhor performance nos dados dos índices financeiros e nos dados que envolvem índices financeiros e cripto-moedas com frequência diária. Como citado anteriormente, a melhor abordagem de extração de variáveis foi a *Feature Matrix*, que utiliza descritores estatísticos para representar as séries.

Capítulo 5

Conclusão

O presente trabalho trouxe novos algoritmos de agrupamento para séries temporais tipo-intervalo considerando distâncias adaptativas e kernelizadas. Dessa forma, buscamos trazer abordagens alternativas aos métodos existentes na literatura. Ademais, utilizamos as mesmas abordagens propostas por Maharaj, Teles, e Brito (2019) para extração de característica das séries e comparamos nossos algoitmos com método K-Means, que utiliza a distância euclidiana

Visando obter um melhor agrupamento, a utilização de novas distâncias trouxe resultados interessantes. A avaliação de desempenho entre os métodos foi realiza considerando dados sintéticos e dados reais, por meio dos índices de Rand Ajustado (dados simulados) e dos índices de Silhueta, Conectividade e Dunn (dados reais).

A utilização das distâncias adaptativas e da aplicação das funções Kernel nas distâncias, permitiu a realização de agrupamentos considerando pesos diferentes entre variáveis bem como aplicar uma métrica de dissimilaridade quando os dados não são linearmente separáveis (funções Kernel).

Antes de aplicar os métodos a dados reais, simulamos séries e aplicamos os métodos, utilizando o índice externo Rand Ajustado, visto que já que conhecíamos a partição *a priori*. Percebemos que a melhor abordagem de extração de características das séries foi o Feature Matrix, porém ainda não foi possível identificar qual método de agrupamento seria o melhor.

Resultados mais conclusivos vieram com a aplicação a dados reais. Os dados escolhidos foram de séries temporais tipo-intervalo de índices financeiros e criptomoedas. Concluímos que o método K-means utilizando as distâncias baseadas em funções kernel trouxeram resultados melhores que método o K-means com distância euclidiana em duas bases de dados analisadas.

Também concluímos que a abordagem de extração de descritores estatísticos das séries temporais de dados tipo intervalo Feature Matrix apresentou os melhores resultados, se comparada com a abordagem STAR.

5.1 Trabalhos Futuros

Acreditamos que há espaço para ampliar esta pesquisa, desenvolvendo novos algoritmos de agrupamento para Séries Temporais tipo-intervalo considerando a kernelização da norma de Frobenius para matriz de autocorrelação, bem como considerando a kernelização das medidas de autocorrelação intervalar. É importante mencionar que, dada a extensão deste trabalho, não foi possível explorar também esta vertente.

Referências Bibliográficas

ABRAHAM, A., 2009, Foundations of computational intelligence. Berlin Heidelberg, Springer. ISBN: 978-3-642-01090-3.

AGHABOZORGI, S., SHIRKHORSHIDI, A. S., WAH, T. Y., 2015, "Time-series clustering – A decade review", Information Systems, , n. 53 (maio), pp. 16–38.

AKAIKE, H., 1974, "A new look at the statistical model identification", IEEE Transactions on Automatic Control, v. 19, n. 6, pp. 716–723. doi: 10.1109/TAC.1974.1100705.

ALI, M., ALQAHTANI, A., JONES, M. W., et al., 2019, "Clustering and Classification for Time Series Data in Visual Analytics: A Survey", IEEE Access, v. 7, pp. 181314–181338. doi: 10.1109/ACCESS.2019.2958551.

ANDERBERG, M., 1973, Cluster analysis for applications. New York, Academic Press. ISBN: 9781483191393.

ARBELAITZ, O., GURRUTXAGA, I., MUGUERZA, J., et al., 2013, "An extensive comparative study of cluster validity indices", Pattern Recognition, , n. 46, pp. 243–256.

ARROYO, J., 2008, Métodos de Predicción para Series Temporales de Intervalos e Histogramas. Tese de Doutorado, Universidad Pontificia Comillas, Madrid.

ARROYO, J., MATÉ, C., 2009, "Forecasting histogram time series with K-nearest neighbours methods", International Journal of Forecasting, v. 25 (03), pp. 192–207.

BILLARD, L., DIDAY, E., 2006, Symbolic data analysis : conceptual statistics and data mining. Chichester, England Hoboken, NJ, John Wiley & Sons Inc. ISBN: 978-0-470-09016-9.

BOX, G., 1970, Time series analysis; forecasting and control. San Francisco, Holden-Day. ISBN: 9780816210947.

BROCKWELL, P. J., DAVIS, R. A., 2002, Introduction to Time Series and Forecasting. 3.

CARVALHO, F., BRITO, P., BOCK, H.-H., 2006, "Dynamic clustering for interval data based on L_2 distance", Computational Statistics, v. 21 (02), pp. 231–250.

CHAVENT, M., LECHEVALLIER, Y., 2002, "Dynamical Clustering of Interval Data: Optimization of an Adequacy Criterion Based on Hausdorff Distance". In: Jajuga, K., Sokołowski, A., Bock, H.-H. (Eds.), Classification, Clustering, and Data Analysis, pp. 53–60, Berlin, Heidelberg. Springer Berlin Heidelberg.

CHIŞ, M., BANERJEE, S., HASSANIEN, A. E., 2009, "Foundations of Computational, Intelligence". v. 6, cap. Clustering Time Series Data: An Evolutionary Approach, Springer, Berlin, Heidelberg.

COSTA, A. F., PIMENTEL, B. A., DE SOUZA, R. M., 2010, "A kernel k-means clustering method for symbolic interval data". In: The 2010 International Joint Conference on Neural Networks (IJCNN), pp. 1–6. doi: 10.1109/IJCNN.2010.5596801.

COWPERTWAIT, P. S., METCALFE, A. V., 2009, Introductory Time Series with R. 1 ed. New York, Springer Science+Business Media.

DUNN, J. C., 1973, "A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters", Journal of Cybernetics, v. 3, n. 3 (sep), pp. 32–57.

FERREIRA, F. A., FERREIRA, D. D., BARBOSA, B. H. G., 2020, "Índice de validação de agrupamento de dados baseado em curvas principais", DOI.

FERREIRA, M. R. P., 2013, Agrupamento Baseado em Kernel com Ponderação Automática das Variáveis via Distâncias Adaptativas. Tese de Doutorado, Universidade Federal de Pernambuco.

FILIPPONE, M., CAMASTRA, F., MASULLI, F., et al., 2008, "A survey of kernel and spectral methods for clustering", ScienceDirect, , n. 41, pp. 176–190.

FOWLKES, E. B., MALLOWS, C. L., 2012, "A Method for Comparing Two Hierarchical Clusterings", Journal of the American Statistical Association, v. 78, n. 383 (mar).

GARCÍA, C., MATÉ, A. C., 2010, "Electric power demand forecasting using interval time series: A comparison between VAR and iMLP", Energy Policy, v. 38, n. 2 (feb), pp. 715–725.

HAMMOUDA, K., KARRAY, F., 2000, "A comparative study of data clustering techniques", Tools of Intelligent Systems Design, v. 5 (01).

HAUTAMAKI, V., NYKANEN, P., FRANTI, P., 2008, "Time-series clustering by approximate prototypes". In: 2008 19th International Conference on Pattern Recognition, pp. 1–4.

JAIN, A. K., 2010, "Data clustering: 50 years beyond K-means", Pattern Recognition Letters, v. 31, n. 8, pp. 651–666. ISSN: 0167-8655. doi: https://doi.org/10.1016/j.patrec.2009.09.011. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0167865509002323>. Award winning papers from the 19th International Conference on Pattern Recognition (ICPR).

JAVED, A., HAMSHAW, S. D., RIZZO, D. M., et al., 2019, "Analysis of Hydrological and Suspended Sediment Events from Mad River Wastershed using Multivariate Time Series Clustering", CoRR, v. abs/1911.12466. Disponível em: <http://arxiv.org/abs/1911.12466>.

JAVED, A., LEE, B. S., RIZZO, D. M., 2020, "A benchmark study on time series clustering", Machine Learning with Applications, (sep). Disponível em: <https://www.sciencedirect.com/science/article/pii/S2666827020300013>.

KALPAKIS, K., GADA, D., PUTTAGUNTA, V., 2001, "Distance measures for effective clustering of ARIMA time-series". In: Proceedings 2001 IEEE International Conference on Data Mining, pp. 273–280.

KAUFMAN, ROUSSEAU, 2005, Finding groups in data : an introduction to cluster analysis. Hoboken, N.J, Wiley. ISBN: 978-0471735786.

KIM, D.-W., LEE, K., LEE, D., et al., 2005, "Evaluation of the performance of clustering algorithms in kernel-induced feature space", Pattern Recognition, , n. 38, pp. 607–611.

KUMAR, M., PATEL, N., WOO, J., 2002, "Clustering seasonality patterns in the presence of errors". pp. 557–563, 01. doi: 10.1145/775107.775129.

MACQUEEN, J., 1967, "Some methods for classification and analysis of multivariate observations". In: In 5-th Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297.

MAHARAJ, E. A., TELES, P., BRITO, P., 2019, "Clustering of interval time series", Stat Comput, , n. 29 (setembro), pp. 1011–1034.

MAIA, A. L. S., CARVALHO, F. A., LUDERMIR, T. B., 2008, "Forecasting models for interval-valued time series", Neurocomputing, , n. 71, pp. 3344–3352.

MERCER, J., 1909, "Functions of Positive and Negative Type, and their Connection with the Theory of Integral Equations", Philosophical Transactions of the Royal Society A, v. 209, pp. 415–446.

RAMANCHARLA, P. K., P.NAGABHUSHAN, 2006, "Time series as a point — a novel approach for time series cluster visualization", Proceedings of the Conference on Data Mining, p. 24–29.

REBBAPRAGADA, U., PROTOPAPAS, P., BRODLEY, C. E., et al., 2009, "Finding anomalous periodic time series", Machine Learning, v. 74 (mar), pp. 281–313.

RIVERA, G. G., ARROYO, J., 2012, "Time series modeling of histogram-valued data: The daily histogram time series of S&P500 intradaily returns", International Journal of Forecasting, v. 28, n. 1 (apr), pp. 20–33.

ROUSSEEUW, P. J., 1987, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis", Journal of Computational and Applied Mathematics, v. 20, pp. 53–65. ISSN: 0377-0427. doi: https://doi.org/10.1016/0377-0427(87)90125-7. Disponível em: <https://www.sciencedirect.com/science/article/pii/0377042787901257>.

SANTOS, J. M., EMBRECHTS, M., 2009, "On the Use of the Adjusted Rand Index as a Metric for Evaluating Supervised Classification". In: International Conference on Artificial Neural Networks, v. 5769, pp. 175–184. Springer, Berlin, Heidelberg.

SCHWARZ, G., 1978, "Estimating the Dimension of a Model", Annals of Statistics, v. 6, n. 2 (march), pp. 461–464.

SCHÖLKOPF, B., SMOLA, A., MÜLLER, K.-R., 1998, "Nonlinear Component Analysis as a Kernel Eigenvalue Problem", Neural Computation, v. 10, n. 5 (jul), pp. 1299 – 1319.

STEINBACH, M., TAN, P.-N., KUMAR, V., et al., 2003, "Discovery of Climate Indices Using Clustering". In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '03, p. 446–455, New York, NY, USA. Association for Computing Machinery. ISBN: 1581137370. doi: 10.1145/956750.956801. Disponível em: <https://doi.org/10.1145/956750.956801>.

SUBHANI, N., RUEDA, L., NGOM, A., et al., 2010, "Multiple gene expression profile alignment for microarray time-series data clustering", Bioinformatics, v. 26, n. 11 (sep), pp. 2281–2288.

VAN DEN HEUVEL, M., MANDL, R., POL, H. H., 2008, "Normalized Cut Group Clustering of Resting-State fMRI Data", Plos One, v. 3, n. 4 (apr).

WARREN LIAO, T., 2005, "Clustering of time series data—a survey", Pattern Recognition, v. 38, n. 11, pp. 1857 – 1874. ISSN: 0031-3203. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0031320305001305>.

XU, D., TIAN, Y., 2015, "A Comprehensive Survey of Clustering Algorithms", Annals of Data Science, v. 2, n. 2 (8), pp. 165–193. doi: 10.1007/s40745-015-0040-1.

XU, R., WUNSCH, D., 2005, "Survey of Clustering Algorithms", IEEE TRANSACTIONS ON NEURAL NETWORKS, v. 16, n. 3 (may).

ZHANG, HAIZHENG, LESSER, et al., 2006, "Multi-Agent Based Peer-to-Peer Information Retrieval Systems with Concurrent Search Sessions". AAMAS '06, p. 305–312, New York, NY, USA. Association for Computing Machinery. ISBN: 1595933034. doi: 10.1145/1160633.1160685. Disponível em: <https://doi.org/10.1145/1160633.1160685>.