
Universidade Federal da Paraíba
Centro de Ciências Exatas e da Natureza
Departamento de Estatística

Kernel K -médias com Distâncias Adaptativas para Dados
Intervalares

José Nataniel Andrade de Sá

Março/2020

José Nataniel Andrade de Sá

Kernel K -médias com Distâncias Adaptativas para Dados
Intervalares

Monografia apresentada ao Curso de Bacharelado em Estatística da Universidade Federal da Paraíba como requisito fundamental para obtenção do Grau de Bacharel em estatística.

Orientador: Prof. Dr. Marcelo Rodrigo Portela Ferreira

João Pessoa
Março de 2020

Catálogo na publicação
Seção de Catalogação e Classificação

S11k Sá, José Nataniel Andrade de.
Kernel K-médias com Distâncias Adaptativas para Dados
Intervalares / José Nataniel Andrade de Sá. - João
Pessoa, 2020.
92 f. : il.

Orientação: Marcelo Rodrigo Portela Ferreira.
Monografia (Graduação) - UFPB/CCEN.

1. Dados Simbólicos. 2. Agrupamento. 3. Ponderação
automática das variáveis. I. Ferreira, Marcelo Rodrigo
Portela. II. Título.

UFPB/CCEN



UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE CIÊNCIAS EXATAS E DA
NATUREZA
COORDENAÇÃO DE ESTATÍSTICA



ATA DE DEFESA DO TRABALHO DE CONCLUSÃO DE CURSO

“Kernel K-médias com Distâncias Adaptativas para Dados Intervalares”

José Nataniel Andrade de Sá

No décimo primeiro dia do mês de março de 2020 às 10:00h na sala CCENA 105 do Centro de Ciências Exatas e da Natureza, a Banca Examinadora do Trabalho de Conclusão de Curso do(a) aluno(a) José Nataniel Andrade de Sá, mat. 2016028169, foi composta pelos professores: Dr. Marcelo Rodrigo Portela Ferreira, Presidente/Orientador(a) (Departamento de Estatística - UFPB), Dr. Eufrásio de Andrade Lima Neto, Examinador(a) (Departamento de Estatística – UFPB), Dr. Telmo de Menezes e Silva Filho, Examinador(a) (Departamento de Estatística – UFPB) e Dra. Juliana Freitas Pires, Examinador(a) Suplente (Departamento de Estatística – UFPB). Dando início aos trabalhos, o presidente da banca cumprimentou os presentes, comunicou aos mesmos a finalidade da reunião e passou a palavra ao candidato para que se fizesse, oralmente, a exposição do trabalho de conclusão de curso intitulado **“Kernel K-médias com Distâncias Adaptativas para Dados Intervalares”**. Concluída a apresentação, o(a) candidato(a) foi arguido(a) pela Banca Examinadora que sugeriu que o(a) aluno(a) fizesse algumas alterações até o dia 1 de abril de 2020. Uma vez entregue a versão final do Trabalho de Conclusão de Curso à Coordenação do Bacharelado em Estatística com as alterações solicitadas pela banca examinadora dentro do prazo que o aluno recebeu, o(a) aluno(a) será aprovado com a nota 10 (DC2), que é a média aritmética das notas atribuídas pelos membros da Banca Examinadora.

Professor(a) Orientador(a) Dr. Marcelo Rodrigo Portela Ferreira

Professor(a) Dr. Eufrásio de Andrade Lima Neto

Professor(a) Dr. Telmo de Menezes e Silva Filho

Aluno(a) José Nataniel Andrade de Sá

João Pessoa, 11 de março de 2020.

Este trabalho é dedicado a ...

*A Deus, por tudo. Aos meus pais, Francilene e José.
A minha tia, Ana. Aos demais familiares e amigos,
que direta ou indiretamente contribuíram para o meu
progresso até aqui. Com amor, vos dedico.*

Agradecimentos

Agradeço primeiramente ao senhor Deus por me conceder tudo que eu precisava para concluir essa etapa da minha vida. Grato pela fé, paciência, calma, perseverança, sabedoria, humildade, ensinamentos e pelas vezes que minha vontade não coincidiu com a do senhor. Sem ti, nada teria acontecido.

Agradeço aos meus pais, Francilene e José, e a titia, Ana. Vocês e suas histórias de vida me motivam segundo a segundo. Muito obrigado pelos conselhos, ensinamentos, dedicação, paciência, patrocínios (rsrsrs), por todo amor e compreensão. Minha gratidão é eterna. Essa conquista é mais de vocês do que minha. Eu amo vocês.

As minhas irmãs, Ana Beatriz e Ana Cláudia, que apesar de toda distância e inocência, sempre se fizeram presente no meu cotidiano;

A minha tia, Joaquina, pelo conselhos, carinho, cuidado (principalmente com as pessoas citadas acima), etc. Você é luz na minha vida.

As minhas avós, Francisca (*in memoriam*) e Maria, obrigado por todos os conselhos, cuidados e amor para comigo. Os ensinamentos de vocês perpetuarão pra sempre em mim. Obrigado por tudo.

Sem citar nomes para não ser injusto, queria agradecer a toda minha família, desde os mais próximos até os mais distantes, por direta ou indiretamente estarem presentes em minha vida. A vocês, minha sincera gratidão.

Um agradecimento especial ao professor Marcelo pela orientação desde os tempos de PIBIC. Obrigado professor, por todos os ensinamentos, conselhos, pela confiança, brincadeiras, pela motivação e pela amizade construída ao longo de quase três anos trabalhando juntos. O senhor é incrível.

Aos professores do DE, por todos os ensinamentos, disponibilidade, conselhos e amizade construída ao longo dessa graduação. Em especial queria agradecer aos professores Hemílio, Ana Flávia, Tarciana, Luíz, Pedro, Cláudio, Eufrásio, Rodrigo, Ulisses, Telmo, Neir e Everlane. Vocês são exemplos de profissionais. Obrigado por tudo.

Mateus, obrigado por tudo cara. Grato por toda parceria, pelos momentos de descontração, pela noites em claro de estudo, por tudo que eu aprendi contigo, etc. Você em muito agregou em minha vida profissional e pessoal. Eduardo, Ullysses e Giovani, muito obrigado por todas as brincadeiras, pelos momentos difíceis, por tudo que vocês me ensinaram e por estarem sempre comigo. Sou grato por ter encontrado vocês nessa caminhada.

A todos os meus amigos de turma, por todos os aprendizados e brincadeiras, em especial queria agradecer a Milleny, Marina, Franciele, Juliana, Gleizi e Kleber. Obrigado por se fazerem presente todo esse tempo, pelas comilanças, pelo apoio e aprendizado. Vocês são para vida.

A todos os amigos e colegas que a estatística me deu. Em especial queria agradecer a Kelfânio, Fernanda, Manuel e Marcos.

As amizades construídas durante o período de estágio no PROCON-JP, em especial ao setor de pesquisa.

Queria agradecer ao meu amigo Zezito, por toda amizade, brincadeiras e por todo apoio, principalmente, nos meses iniciais do curso.

Um agradecimento especial a minha turma do apartamento 306: Milleny, Zé, Samara e João. Muito obrigado por tudo, pelas brincadeiras, pelo companheirismo, compreensão e por serem minha família aqui em João Pessoa. Sou grato demais por vocês.

Aos meus amigos (as) de minha querida “sanzé”. Obrigado por toda amizade e por todos os momentos juntos. Apesar de toda a distância, vocês tiveram papel fundamental nessa conquista. Obrigado por tudo.

Aos professores Eufrásio e Telmo, pela disposição em participar da banca avaliadora e por todas as sugestões e incentivos a esse trabalho.

Ao CNPQ por todo apoio financeiro durante a graduação.

“Provai e vede como Deus é bom, feliz de quem no seu amor confia”.

— *Bíblia Sagrada, Salmos 34, 9*

“Não fostes vós que me escolhestes; fui eu que vos escolhi ”.

— *Bíblia Sagrada, Jó 15, 16*

“Tentou. Falhou. Não importa. Tente de novo. Falhe de novo. Falhe melhor... Você erra 100% dos tiros que não dá ”.

— *Samuel Beckett e Wayne Gretzky*

A Análise de Dados Simbólicos (ADS) ou *Symbolic Data Analysis (SDA)* é uma extensão da análise de dados clássicos que objetiva desenvolver técnicas estatísticas e/ou computacionais para dados mais complexos que os dados usuais, geralmente representados por valores e categorias, podendo incluir variações e estruturas internas. Diversos métodos de agrupamentos para dados clássicos já foram estendidos para os dados simbólicos, como por exemplo o K-médias e suas versões kernelizadas. Funções de *kernel* tem sido amplamente utilizadas em algoritmos de agrupamento, devido à melhora do desempenho desses métodos em situações nas quais os grupos não são linearmente separáveis. Ainda assim, esses métodos baseados em *kernel* consideram que as variáveis são igualmente importantes para o processo de agrupamento, o que na maioria das vezes não se configura desta forma, principalmente na era do *Big Data* em que conjuntos de dados com alta dimensão são facilmente encontrados. O objetivo deste trabalho é propor novos métodos de agrupamentos para dados simbólicos dos tipo intervalo baseado em *kernel* com ponderação automática das variáveis via distâncias adaptativas, que mudam a cada iteração do algoritmo e são obtidas como soma de distâncias euclidianas quadradas entre as observações e os centroides de cada grupo, calculadas para cada variável de forma individual. A principal vantagem dos métodos propostos sobre a abordagem convencional é que o uso de distâncias adaptativas permite atribuir pesos às variáveis, possibilitando diferenciar a importância das mesmas e, conseqüentemente, melhorar o desempenho do algoritmo. Este trabalho engloba o paradigma de agrupamento rígido (*hard*) e considera duas vertentes usadas em abordagem de *kernel*, onde uma considera que os protótipos estão definidos no espaço original dos dados e outra considera que os protótipos estão definidos em um espaço de maior dimensão, denominado espaço de características. Experimentos realizados com dados simulados e dados reais intervalares mostram a eficiência dos métodos propostos.

Palavras-chave: Dados Simbólicos; Agrupamento; Ponderação automática das variáveis.

Symbolic Data Analysis (SDA) is an extension of classic data analysis that aims to develop statistical and/or computational techniques for more complex data than the usual data, usually represented by values and categories, may include variations and internal structures. Several clustering methods for classical data have already been extended for symbolic data, such as K-means and its kernelized versions. The kernel functions have been widely used in clustering algorithms due to improvements in their performance in scenarios where the clusters are non-linearly separable. Even so, these kernel-based methods consider that variables are equally important in the clustering process, which most of the time is not configured in this way, especially in the era of Big Data, where high-dimensional data sets are easily found. The objective of this work is to propose new kernel-based clustering methods for interval-valued symbolic data with automatic weighting of variables through adaptive distances that change at each iteration of the algorithm and are obtained as the sum of the square Euclidean distances between the observations and centroids of each group, calculated for each variable, individually. The main advantage of the proposed methods over the conventional approach is the use of adaptive distances which allows to assign weights to variables, making possible to differentiate the importance of the variables and consequently improve the performance of the algorithms. This work encompasses the hard clustering paradigm and considers two aspects used in kernel functions, when the prototypes are defined in the original data space and when the prototypes are defined in a higher dimension space, known as feature space. Experiments with simulated data and real interval-valued data shows the efficiency of the proposed methods.

Keywords: Symbolic Data; Clustering; Automatic variable weighting.

Lista de Algoritmos	viii
Lista de Tabelas	x
1 Introdução	1
1.1 Introdução	1
1.2 Organização do trabalho	5
2 Métodos de agrupamento <i>hard</i> para dados convencionais	6
2.1 Método K -médias	7
2.2 Conceitos básicos de <i>kernel</i>	8
2.3 Método <i>kernel</i> K -médias baseado na kernelização da métrica	10
2.4 Método <i>kernel</i> K -médias no espaço de características	12
3 Métodos de agrupamento para dados simbólicos do tipo intervalo	15
3.1 Introdução	15
3.2 Tipos de variáveis simbólicas	17
3.3 Funções de <i>kernel</i> para dados simbólicos do tipo intervalo	18
3.4 Métodos de agrupamentos para dados simbólicos do tipo intervalo	19
3.4.1 Método K -médias para intervalos	20
3.4.2 <i>Kernel</i> K -médias para dados do tipo intervalo baseado na kerneli- zação da métrica	21
3.4.3 <i>Kernel</i> K -médias para dados do tipo intervalo no espaço de carac- terísticas	23
4 Agrupamento <i>hard</i> baseado em <i>kernel</i> com ponderação automática das variáveis para dados intervalares	26

4.1	Introdução	26
4.2	Agrupamento <i>hard</i> baseado na kernelização da métrica com ponderação automática das variáveis para dados do tipo intervalo	31
4.3	Agrupamento <i>hard</i> no espaço de características com ponderação automática das variáveis para dados do tipo intervalo	43
4.4	Convergência dos métodos de agrupamento <i>hard</i> baseado em <i>kernel</i> com ponderação automática das variáveis para dados do tipo intervalo	50
5	Avaliação Experimental	53
5.1	Introdução	53
5.1.1	Conjuntos de dados simulados do tipo intervalo	53
5.1.2	Conjuntos de dados reais do tipo intervalo	57
5.2	Índices de avaliação	61
5.3	Resultados	63
5.3.1	Conjunto de dados simulados do tipo intervalos	64
5.3.2	Conjunto de dados reais do tipo intervalo	68
6	Conclusões	74
6.1	Trabalhos futuros	76
	Referências bibliográficas	77

Lista de Algoritmos

1	MÉTODO K -MÉDIAS	8
2	MÉTODO <i>kernel</i> K -MÉDIAS BASEADO NA KERNELIZAÇÃO DA MÉTRICA . .	12
3	MÉTODO <i>kernel</i> K -MÉDIAS NO ESPAÇO DE CARACTERÍSTICAS	14
4	MÉTODO K -MÉDIAS PARA INTERVALOS	21
5	<i>Kernel</i> K -MÉDIAS PARA DADOS DO TIPO INTERVALO BASEADO NA KERNELIZAÇÃO DA MÉTRICA	23
6	<i>Kernel</i> K -MÉDIAS PARA DADOS DO TIPO INTERVALO NO ESPAÇO DE CARACTERÍSTICAS	25
7	ALGORITMO PARA A OBTENÇÃO DO AGRUPAMENTO PELOS MÉTODOS PROPOSTOS NA KERNELIZAÇÃO DA MÉTRICA	42
8	ALGORITMO PARA A OBTENÇÃO DO AGRUPAMENTO PELOS MÉTODOS PROPOSTOS NO ESPAÇO DE CARACTERÍSTICAS	49

Lista de Tabelas

2.1	Exemplos de funções <i>kernel</i>	10
3.1	Tabela de dados clássicos	16
3.2	Tabela de dados simbólicos referente a descrição de y_k	17
5.1	Informações das estruturas de média e variância do conjunto de dados simulados 1.	54
5.2	Informações das estruturas de média e variância do conjunto de dados simulados 2.	56
5.3	Dimensões dos conjuntos de dados reais.	57
5.4	Matriz de confusão	62
5.5	Resultado dos algoritmos de agrupamento segundo os índices CR e OERC para o conjunto de dados simulados 1 : média e desvio padrão (entre parênteses).	65
5.6	Resultado dos algoritmos de agrupamento segundo os índices CR e OERC para o conjunto de dados simulados 2 : média e desvio padrão (entre parênteses).	66
5.7	Resultado do teste de Wilcoxon pareado nas amostras do conjunto de dados simulados 1 para comparar os métodos segundo as funções <i>kernel</i>	68
5.8	Resultado do teste de Wilcoxon pareado nas amostras do conjunto de dados simulados 2 para comparar os métodos segundo as funções <i>kernel</i>	68
5.9	Resultado do Índice Corrigido de Rand (CR) para os dados simbólicos intervalares reais considerados.	70
5.10	Resultado da Taxa Total de Erro de Classificação (OERC) para os dados simbólicos intervalares reais considerados.	71

5.11	Classificação de desempenho médio e mediano dos algoritmos de agrupamentos de acordo com o Índice Corrigido de Rand (CR) e a Taxa Total de Erro de Classificação (OERC).	72
5.12	Média do desempenho médio e mediano segundo as funções <i>kernels</i> de uma (1 C) e duas (2 C) componentes.	72

1.1 Introdução

Desde o surgimento dos sistemas de computadores impulsionados pelo desenvolvimento tecnológico no século passado, a capacidade de armazenamento de dados aumentou de forma exponencial, possibilitando a origem de bases de dados gigantescas, tanto em tamanho quanto em dimensionalidade. Em paralelo a isso, a ideia de mineração de dados, ou *data mining*, ganhou destaque por sumarizar tais bases, buscando informações úteis para a tomada de decisão. Refere-se então a um campo interdisciplinar, com aplicações em diversas áreas, como: taxonomia, biologia, marketing, medicina e etc, baseado em conceitos estatísticos e computacionais (OLIVEIRA et al., 2018).

Levando em consideração que a forma tradicional de representação de um base de dados é uma matriz $n \times p$, onde n representa o número de observações e p o número de variáveis estudadas, quando se trata de um conjunto de dados gigantesco, os valores de n e p podem ser da ordem de milhões e milhares, respectivamente, gerando problemas de custo computacional, como por exemplo, em operações matriciais. Segundo COSTA (2011, p.3) “em virtude desse crescente volume de dados, os métodos tradicionais de análise de dados têm se tornado inapropriados, pois não conseguem analisar o conteúdo das informações com a finalidade de obter conhecimentos importantes”.

Em muitas aplicações, faz-se necessário levar em consideração a precisão e variabilidade dos dados para representar a informação disponível. Considere, por exemplo, o caso em

que dois pacientes tem suas taxas glicêmicas acompanhadas durante um certo período de tempo. Um paciente considerado sem diabetes pode apresentar taxas variando no intervalo [85,90], enquanto outro paciente também sem diabetes pode apresentar taxas entre [90,98]. Nesse caso, uma análise através do ponto médio dos intervalos perderia a informação sobre variação da glicemia de cada paciente. Nuances como essa não são captadas por meio dos dados clássicos, sendo necessário uma nova representação de dados.

Uma alternativa é usar a ideia de dados simbólicos, que podem ser representados por listas, intervalos, histogramas e afins. Dados desse tipo são adequados para tratar de dados provenientes de imprecisões ou valores estimados via intervalo de confiança, limites de valores de um item e variações de uma variável, seja no tempo ou por reduções em grupos, sendo interesse de estudo, principalmente, da Análise de Dados Simbólicos (ADS) – Symbolic Data Analysis (SDA) – que constitui uma nova área de pesquisa relacionada à análise multivariada, reconhecimento de padrões e inteligência artificial. O objetivo principal da Análise de Dados Simbólicos é estender os métodos existentes para dados clássicos, como também desenvolver novos métodos para a análise de dados mais complexos que os usuais, que podem ser representados por conjuntos de categorias, intervalos, distribuições de frequência, histogramas, distribuições de probabilidade, etc (BILLARD; DIDAY, 2003; BOCK; DIDAY, 2000).

Métodos de agrupamento são ferramentas usadas na mineração de dados e tem sido amplamente utilizados para o reconhecimento não supervisionado de padrões. Áreas como processamento de imagem e recuperação de informação são exemplos de onde se utilizam estes métodos, que por sua vez tem diversas aplicações práticas, como, por exemplo, segmentação de imagem e segmentação de mercado. A análise de agrupamento é uma técnica multivariada que busca encontrar grupos homogêneos a partir de um conjunto de indivíduos (padrões, objetos e etc) (MURTY; JAIN; FLYNN, 1999). Mais especificamente, os métodos de agrupamento formam os grupos de modo a se obter homogeneidade em cada grupo e heterogeneidade entre eles, ou seja, observações pertencentes a um mesmo grupo apresentem um alto grau de similaridade, enquanto observações pertencentes a grupos distintos tenham alto grau de dissimilaridade .

Os métodos de agrupamento mais populares são divididos em: hierárquicos e particionais. Os métodos hierárquicos fornecem uma sequência de dados aninhados, resultando em uma estrutura de dependência entre os grupos, onde geralmente são representados por um dendrograma. Em contrapartida, os métodos particionais fornecem uma única partição dos dados de entrada em determinado número de grupos, por meio da otimização de uma função objetivo, produzindo hipersuperfícies de separação entre os grupos. Os métodos particionais por sua vez são desenvolvidos sob duas matrizes: agrupamento rígido (*hard*) (JAIN, 2010) e agrupamento difuso (*fuzzy*) (HÖPPNER et al., 1999). No

agrupamento rígido, os grupos são disjuntos e cada observação (objeto) pertence a um, e somente um, grupo. Por outro lado, no agrupamento difuso, as observações podem pertencer a todos os grupos com um certo grau de pertinência.

Uma componente importante para qualquer método de agrupamento é a medida de similaridade, e medidas de distâncias são exemplos destas, na qual a distância euclidiana é a mais comumente utilizada. Os métodos convencionais que usam a distância euclidiana como medida de similaridade fornecem bons resultados quando usados em bancos de dados onde os grupos são, aproximadamente, linearmente separáveis e hipersféricos. Contudo, quando o conjunto de dados não segue esse critério, isto é, a estrutura dos dados é complexa, a performance desses métodos de agrupamento pode ser insatisfatória. O método K -médias desenvolvido por Steinhaus (1956) – o método particional mais conhecido e amplamente utilizado – é exemplo dessa situação. Esse algoritmo fornece uma partição dos dados em K grupos, onde cada grupo é caracterizado por um ponto central conhecido com centróide ou protótipo. Outra limitação desse método é que ele encontra ótimos locais, ficando sensível a escolhas dos protótipos iniciais (JAIN, 2010).

Nesse sentido, vários métodos capazes de tratar de dados com estrutura complexa estão sendo desenvolvidos, dentre os quais, métodos baseados em funções de *kernel* tem ganhado destaque. A natureza desses métodos está baseada na realização de um mapeamento não-linear arbitrário Φ do espaço original de dimensão p ($X \subset \mathbb{R}^p$) para um espaço de dimensão q (possivelmente infinito), denominado espaço de características, \mathcal{F} , onde $q > p$. O motivo para realizar esse mapeamento se deve ao fato que os grupos, que até então apresentavam uma estrutura complexa, passam a ser bem definidos e linearmente separáveis. A grande vantagem de tais métodos é que produtos internos no espaço de características podem ser calculadas através de um *kernel*, denominado *kernel* de Mercer, \mathcal{K} , dado por $\mathcal{K}(\mathbf{x}, \mathbf{x}') = \Phi(\mathbf{x})^\top \Phi(\mathbf{x}')$, onde $\mathbf{x}, \mathbf{x}' \in X, X \subset \mathbb{R}^p$ (MERCER, 1909).

A partir do desenvolvimento do algoritmo *kernel* K -médias (GIROLAMI, 2002) diversos métodos de agrupamento foram modificados de modo a incorporarem funções *kernel*. Além disso, uma grande variedade de métodos de agrupamento baseados em *kernel* têm sido propostos (FILIPPONE et al., 2008). Esses novos métodos passaram a considerar duas abordagens: kernelização da métrica, onde os centroides são obtidos no espaço original de entrada e a distância dos padrões aos centroides são calculadas via função *kernel*, e agrupamento no espaço de característica, onde os protótipos são obtidos de forma implícita no espaço de característica.

Diversos algoritmos de agrupamento para dados simbólicos tem sido propostos, e muitos deles baseados em *kernel*. Costa, Pimentel e Souza (2010) propuseram o método *kernel* K -médias e o *kernel* K -médias *difuso* para dados intervalares, onde cada padrão é des-

critico por um vetor de intervalos. Oliveira et al. (2018) propuseram um algoritmo do tipo subtrativo para dados intervalares baseado no trabalho de Kim et al. (2005). Carvalho, Brito e Bock (2006) propuseram o algoritmo K -médias para dados intervalares baseado na distância L2. No entanto, apesar dos muitos trabalhos desenvolvidos, ainda há uma lacuna a ser preenchida por esses métodos baseados em *kernel*.

Em análise de agrupamento para dados simbólicos do tipo intervalo os padrões são comumente representados por um vetor, onde cada componente deste é um intervalo referente a uma medição de uma variável. Nos algoritmos convencionais, bem como nas suas versões kernelizadas, as variáveis são consideradas como sendo igualmente importantes, do ponto de vista de que todas possuem o mesmo peso para o processo de agrupamento, quando na verdade, em diversas áreas do conhecimento – em especial, para conjuntos de dados de alta dimensão – podem existir variáveis irrelevantes, como também variáveis mais importantes do que outras, e ainda, o conjunto de variáveis importantes para cada grupo pode ser diferente.

Motivados pelo trabalho de FERREIRA (2013), onde foi considerado ponderação automática das variáveis para dados convencionais, nesse trabalho, propomos métodos de agrupamento baseado em *kernel* para dados simbólicos do tipo intervalo com ponderação automática das variáveis por meio de distâncias adaptativas, na qual a medida de similaridade é obtida através de soma de distâncias euclidianas entre as observações e os centroides, calculada para cada variável, individualmente, via funções *kernel*. A grande vantagem dessa abordagem sobre os algoritmos convencionais de agrupamento baseados em *kernel* consiste na possibilidade de trabalhar com distâncias que mudam a cada iteração do algoritmo (distâncias adaptativas) e podem ser a mesma para todos os grupos (distâncias adaptativas globais) ou diferentes de um grupo para outro (distâncias adaptativas locais). FERREIRA (2013) afirma que: “Em algumas situações, distâncias adaptativas locais podem não ser apropriadas porque podem levar o algoritmo a pontos de mínimos locais produzindo assim, soluções sub-ótimas”. Por esta razão, neste trabalho consideramos ambos os tipos de distâncias (global e local) para a construção dos métodos de agrupamento. Outrossim, os métodos propostos aqui foram desenvolvidos sobre duas vertentes: abordagem na kernelização da métrica e abordagem no espaço de característica. Por fim, utilizamos como função *kernel* duas versões do *kernel* gaussiano, com uma e duas componentes, e para a derivação das expressões matemáticas consideramos duas restrições: a primeira, de que a soma dos pesos das variáveis deve ser igual a um, e a segunda de que o produto dos pesos das variáveis deve ser igual a um.

1.2 Organização do trabalho

Além deste capítulo de introdução, este trabalho é composto por mais cinco capítulos, como segue:

- No Capítulo 2 apresentamos uma breve revisão acerca da teoria básica dos métodos de agrupamento rígido (*hard*), bem como a ideia dos métodos baseados em *kernel* ;
- No Capítulo 3 introduzimos a definição de dados simbólicos, bem como dissertamos acerca de agrupamentos convencionais para dados simbólico do tipo intervalo, além de apresentar a ideia do *kernel* para esse tipo de dado, e dos métodos baseados em *kernel* para esse caso;
- O Capítulo 4 apresenta a principal contribuição deste trabalho: os métodos de agrupamento baseados em *kernel* com ponderação automática das variáveis via distâncias adaptativas para dados simbólicos do tipo intervalo. Aqui consideramos ambas vertentes: kernelização da métrica e agrupamento no espaço de características. São demonstradas todas as expressões matemáticas obtidas e também é apresentado os algoritmos, para fins computacionais;
- No Capítulo 5 apresentamos alguns índices de avaliação para comparação dos métodos, além de trazer um conjunto de experimentos numéricos aplicados tanto a dados reais quanto a dados simulados. Os resultados obtidos mostram a superioridade dos métodos propostos neste trabalho, em relação aos métodos de agrupamento convencionais ;
- Por fim, no Capítulo 6, apresentamos as conclusões deste trabalho e apontamos direções para trabalhos futuros.

Métodos de agrupamento *hard* para dados convencionais

Os métodos de agrupamento são ferramentas computacionais que buscam sumarizar um conjunto de dados no sentido de separar os objetos em grupos, baseado em suas características. Como citado no capítulo anterior, os métodos mais populares são divididos em hierárquicos e particionais, de modo que os métodos hierárquicos fornecem uma base de hierarquia entre as partições, representadas geralmente por uma estrutura semelhante às árvores denominada dendrograma, enquanto que no agrupamento particional o resultado é uma partição única do conjunto de dados em um número fixo de grupos. Estes últimos buscam obter a partição através da otimização de uma função objetivo que se baseia em uma medida de similaridade, na qual medidas de distância são tipicamente utilizadas. Os métodos particionais são divididos em rígido (*hard*) e difuso (*fuzzy*). No agrupamento *hard* um objeto pertence a um, e somente um, grupo, enquanto que nos métodos *fuzzy* as observações podem pertencer a todos os grupos, com um certo grau de pertinência. A seguir nós apresentamos uma breve revisão acerca do principal método de agrupamento *hard*, o método K -médias, e suas versões kernelizadas, tanto na kernelização da métrica como no espaço de características.

2.1 Método K-médias

Este método foi proposto por Steinhaus (1956) e é o mais famoso e amplamente utilizado (MACQUEEN et al., 1967) devido ao fato de ser de fácil implementação e de baixo custo computacional, em virtude de que sua complexidade é da ordem $O(nKl)$, sendo n , K e l o número de observações, o número de grupos e o número de iterações, respectivamente (CHOUDHARI et al., 2005).

Seja $\Omega = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ um conjunto de n observações, onde cada \mathbf{x}_i ($i = 1, \dots, n$) é descrito por p variáveis e seja $\mathbf{P} = \{P_1, \dots, P_K\}$ uma partição de Ω em K grupos disjuntos, onde cada grupo k ($k = 1, \dots, K$) é caracterizado por um ponto central \mathbf{v}_k , conhecido como protótipo ou centroide. Dessa forma, o K-médias, baseado na distância euclidiana, busca obter os grupos através da minimização da seguinte função objetivo:

$$J^{(t)} = \sum_{k=1}^K \sum_{i \in P_k^{(t)}} \|\mathbf{x}_i - \mathbf{v}_k^{(t)}\|^2, \quad (2.1)$$

onde t refere -se a t -ésima iteração do algoritmo.

É fácil provar que o protótipo \mathbf{v}_k que minimiza J é a média aritmética dos padrões \mathbf{x}_i pertencentes ao grupo k , ou seja:

$$\mathbf{v}_k^{(t+1)} = \frac{1}{n_k} \sum_{i \in P_k^{(t)}} \mathbf{x}_i, \quad (2.2)$$

onde n_k é o número de observações pertencentes ao grupo k .

O algoritmo a seguir resume o passo a passo iterativo para obter o agrupamento pelo método K-médias.

Algoritmo 1: MÉTODO K -MÉDIAS

(1) Inicialização

Fixe K (o número de grupos), $2 \leq K < n$; Escolha K padrões distintos $\mathbf{v}_1, \dots, \mathbf{v}_K$ pertencentes a Ω como protótipos iniciais e aloque cada padrão i de acordo com o protótipo mais próximo \mathbf{v}_h ($h = \arg \min_{1 \leq k \leq K} \|\mathbf{x}_i - \mathbf{v}_k\|^2$) para obter a partição inicial $\mathbf{P} = \{P_1, \dots, P_K\}$;

(2) Atualização dos protótipos

Atualize os protótipos dos grupos \mathbf{v}_k ($k = 1, \dots, K$) segundo a equação 2.2 ;

(3) Atualização da partição

$test \leftarrow 0$

para $i = 1$ até n faça

 defina o grupo vencedor P_h tal que

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

 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).

2.2 Conceitos básicos de *kernel*

O método K -médias apresenta algumas deficiências, podendo-se destacar duas : o algoritmo pode levar a mínimos locais, ficando sensível a escolha dos protótipos iniciais, e a segunda, é que o método só fornece bons resultados quando o conjunto de dados é, aproximadamente, linearmente separável e hiperesférico.

Em relação a primeira limitação algumas propostas surgiram para a escolha inicial dos centroides. Por exemplo, Cui e Potok (2005) e Abraham, Das e Konar (2006) propuseram estratégias que mesclavam as funcionalidades do K -médias e outros algoritmos híbridos. Ainda nesse ponto, Arthur e Vassilvitskii (2006) propuseram o algoritmo K -means++ , na qual eles abordam uma forma estratégica de selecionar os protótipos iniciais do algoritmo.

Nesse trabalho eles mostraram a eficiência do K -means++ em relação ao k -médias tradicional. Quanto a segunda limitação, estratégias como algoritmos de agrupamento espectral, algoritmos de particionamento de grafos e o uso de funções *kernel*, tem sido propostas, de modo que muitos pesquisadores tem demonstrado interesse em desenvolver algoritmos de agrupamento baseado em *kernel* (BEN-HUR et al., 2001; FILIPPONE et al., 2008). A principal motivação para o uso desses métodos é a possibilidade da realização de um mapeamento Φ , não-linear, do espaço original dos dados para um espaço de dimensão mais alta, denominado espaço de características, \mathcal{F} .

Seja $X = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ um conjunto de padrões não-vazio no espaço p -dimensional \mathbb{R}^p . Uma função $\mathcal{K}: X \times X \rightarrow \mathbb{R}$ é dito ser um *kernel* positivo-definido se, e somente se, satisfaz as seguintes condições (MERCER, 1909):

i) \mathcal{K} é simétrica, isto é, $\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) = \mathcal{K}(\mathbf{x}_j, \mathbf{x}_i)$;

ii)

$$\sum_{i=1}^n \sum_{j=1}^n c_i c_j \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) \geq 0, \quad \forall n \geq 2 \quad (2.3)$$

onde, $c_r \in \mathbb{R} \quad \forall r = 1, \dots, n$.

O motivo para o uso do *kernel* se dá através do teorema de Cover (HAYKIN, 1994). O teorema garante que um conjunto de dados não-linear pode ser transformado em um outro espaço, denominado espaço de características, \mathcal{F} , com grandes chances dos dados serem linearmente separáveis, desde que a transformação seja não-linear e que a dimensão de \mathcal{F} seja suficientemente alta (possivelmente infinita).

Considere $\Phi: X \rightarrow \mathcal{F}$ 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 $\mathbf{x}_i^\top \mathbf{x}_j$ é mapeado no espaço de característica como $\Phi(\mathbf{x}_i)^\top \Phi(\mathbf{x}_j)$. 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 propostas por Mercer (1909) pode ser escrito como :

$$\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i)^\top \Phi(\mathbf{x}_j) \quad (2.4)$$

que é conhecida como *kernel trick* (MULLER et al., 2001; SCHÖLKOPF; SMOLA; MÜLLER, 1998).

Usualmente, os métodos de agrupamento baseados em *kernel* trabalham com uma matriz $\mathbf{K} \in \mathbb{R}^{n \times n}$, denominada matriz *kernel*, de modo que $k_{ij} = \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$, $i, j = 1, \dots, n$.

O cálculo de distâncias euclidianas em \mathcal{F} pode ser feito através de *kernel*, devido a forma dada na equação 2.4 (MULLER et al., 2001; SCHÖLKOPF; SMOLA; MÜLLER, 1998), como :

$$\begin{aligned} \|\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j)\|^2 &= [\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j)]^\top [\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j)] \\ &= \Phi(\mathbf{x}_i)^\top \Phi(\mathbf{x}_i) - 2\Phi(\mathbf{x}_i)^\top \Phi(\mathbf{x}_j) + \Phi(\mathbf{x}_j)^\top \Phi(\mathbf{x}_j) \\ &= \mathcal{K}(\mathbf{x}_i, \mathbf{x}_i) - 2\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) + \mathcal{K}(\mathbf{x}_j, \mathbf{x}_j) \end{aligned} \quad (2.5)$$

Alguns exemplos de funções de *kernel* estão listados na tabela a seguir.

Tabela 2.1: Exemplos de funções *kernel*

<i>kernel</i>	Expressão	Restrições dos parâmetros
Gaussiano	$\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(\frac{-\ \mathbf{x}_i - \mathbf{x}_j\ ^2}{2\sigma^2}\right)$	$\sigma > 0$
Polinomial	$\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) = (\lambda \mathbf{x}_i^\top \mathbf{x}_j + \theta)^d$	$\lambda > 0, \theta \geq 0, d \in \mathbb{N}$
Linear	$\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^\top \mathbf{x}_j$	—
Laplaciano	$\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\lambda \ \mathbf{x}_i - \mathbf{x}_j\)$	$\lambda > 0$
Sigmoidal	$\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\lambda \mathbf{x}_i^\top \mathbf{x}_j + \theta)$	$\lambda > 0, \theta \geq 0,$

Como já citado anteriormente, os métodos de agrupamento baseados em *kernel* seguem duas vertentes: kernelização da métrica e agrupamento no espaço de características.

2.3 Método *kernel* K -médias baseado na kernelização da métrica

Seja $\Omega = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ um conjunto de n observações, onde cada \mathbf{x}_i ($i = 1, \dots, n$) é descrito por p variáveis e seja $\mathbf{P} = \{P_1, \dots, P_K\}$ uma partição de Ω em K grupos disjuntos. A ideia do *kernel* K -médias é minimizar a seguinte função objetivo

$$\begin{aligned} J^{(t)} &= \sum_{k=1}^K \sum_{i \in P_k^{(t)}} \|\Phi(\mathbf{x}_i) - \Phi(\mathbf{v}_k^{(t)})\|^2 \\ &= \sum_{k=1}^K \sum_{i \in P_k^{(t)}} \mathcal{K}(\mathbf{x}_i, \mathbf{x}_i) - 2\mathcal{K}(\mathbf{x}_i, \mathbf{v}_k^{(t)}) + \mathcal{K}(\mathbf{v}_k^{(t)}, \mathbf{v}_k^{(t)}) \end{aligned} \quad (2.6)$$

, onde $\mathbf{v}_k^{(t)} \in \mathbb{R}^p$ é o protótipo do k -ésimo grupo ($k = 1, \dots, K$).

Para obtenção do protótipo do k -ésimo grupo devemos calcular a derivada de J em relação a \mathbf{v}_k e para isso é necessário a escolha de um função de *kernel*. Se considerarmos o *kernel Gaussiano*, que é o mais usado na literatura devido a sua facilidade de tratamento analítico (FERREIRA, 2013), então $\mathcal{K}(\mathbf{x}_r, \mathbf{x}_r) = 1, \forall r$, e dessa forma a função objetivo J , dada pela equação (2.6), passa a ser:

$$J^{(t)} = 2 \sum_{k=1}^K \sum_{i \in P_k^{(t)}} \left[1 - \mathcal{K}(\mathbf{x}_i, \mathbf{v}_k^{(t)}) \right] \quad (2.7)$$

Ao realizar a derivação da equação (2.7) em relação a \mathbf{v}_k e igualar ao vetor nulo p -dimensional, obtemos que o valor de \mathbf{v}_k que minimiza J é a média das observações pertencentes ao grupo k ponderada pelos *kernels* entre essas observações e o protótipo do grupo k , ou seja:

$$\mathbf{v}_k^{(t+1)} = \frac{\sum_{i \in P_k^{(t)}} \mathcal{K}(\mathbf{x}_i, \mathbf{v}_k^{(t)}) \mathbf{x}_i}{\sum_{i \in P_k^{(t)}} \mathcal{K}(\mathbf{x}_i, \mathbf{v}_k^{(t)})} \quad (2.8)$$

Após a obtenção dos protótipos \mathbf{v}_k ($k = 1, \dots, K$) o problema agora é encontrar a partição $\mathbf{P} = \{P_1, \dots, P_K\}$ que minimiza J . Como os protótipos \mathbf{v}_k ($k = 1, \dots, K$) estão fixos, a partição $\mathbf{P} = \{P_1, \dots, P_K\}$ que minimiza a função objetivo J , dada pela equação (2.7), é atualizada de acordo com a seguinte regra:

$$P_k^{(t)} = \left\{ i \in \Omega : \|\Phi(\mathbf{x}_i) - \Phi(\mathbf{v}_k^{(t)})\|^2 \leq \|\Phi(\mathbf{x}_i) - \Phi(\mathbf{v}_h^{(t)})\|^2, \forall h \neq k, h = 1, \dots, K \right\} \quad (2.9)$$

O algoritmo a seguir resume o passo a passo iterativo para obter o agrupamento pelo método *kernel* K -médias baseado em kernelização da métrica.

Algoritmo 2: MÉTODO *kernel* K -MÉDIAS BASEADO NA KERNELIZAÇÃO DA MÉTRICA

(1) Inicialização

Fixe K (o número de grupos), $2 \leq K < n$; Escolha aleatoriamente uma partição \mathbf{P} de Ω em K grupos P_1, P_2, \dots, P_K ou, alternativamente, escolha K padrões distintos $\mathbf{v}_1, \dots, \mathbf{v}_K$ pertencentes a Ω como protótipos iniciais e aloque cada padrão i de acordo com o protótipo mais próximo \mathbf{v}_h ($h = \arg \min_{1 \leq k \leq K} \|\Phi(\mathbf{x}_i) - \Phi(\mathbf{v}_k)\|^2$) para obter a partição inicial $\mathbf{P} = \{P_1, \dots, P_K\}$;

(2) Atualização dos protótipos

Atualize os protótipos dos grupos \mathbf{v}_k ($k = 1, \dots, K$) segundo a equação (2.8) ;

(3) Atualização da partição

$test \leftarrow 0$

para $i = 1$ até n faça

defina o grupo vencedor P_h tal que

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

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).

2.4 Método *kernel* K -médias no espaço de características

A ideia central do método *kernel* K -médias no espaço de características é minimizar a função objetivo J para a obtenção de uma partição $\mathbf{P} = \{P_1, \dots, P_K\}$ de Ω em K grupos disjuntos, onde J é calculada da seguinte forma: (CHIANG; HAO, 2003; FILIPPONE et al., 2008; GRAEPEL; OBERMAYER, 1998; CHEN; ZHANG, 2002):

$$J^{(t)} = \sum_{k=1}^K \sum_{i \in P_k^{(t)}} \|\Phi(\mathbf{x}_i) - (\mathbf{v}_k^\Phi)^{(t)}\|^2, \quad (2.10)$$

onde \mathbf{v}_k^Φ é o centroide do k -ésimo grupo no espaço de características.

De forma análoga ao método *kernel* K -médias baseado em kernelização da métrica, é necessário encontrar os centroides \mathbf{v}_k^Φ ($k = 1, \dots, K$) que minimizam a função objetivo J dada pela equação (2.10). Ao minimizar J em relação a \mathbf{v}_k^Φ obtemos a seguinte equação para os centroides dos grupos no espaço de características (FILIPPONE et al., 2008):

$$(\mathbf{v}_k^\Phi)^{(t+1)} = \frac{1}{|P_k^{(t)}|} \sum_{i \in P_k^{(t)}} \Phi(\mathbf{x}_i) \quad (2.11)$$

Como a transformação não-linear Φ não é conhecida de forma explícita, os protótipos no espaço de características \mathbf{v}_k^Φ ($k = 1, \dots, K$) não podem ser obtidos através da equação (2.11). No entanto, a distância $\|\Phi(\mathbf{x}_i) - \mathbf{v}_k^\Phi\|^2$ no espaço de características pode ser obtida, através de funções *kernel* no espaço original dos padrões, da seguinte forma (FERREIRA, 2013):

$$\|\Phi(\mathbf{x}_i) - \mathbf{v}_k^\Phi\|^2 = \mathcal{K}(\mathbf{x}_i, \mathbf{x}_i) - 2 \frac{\sum_{l \in P_k} \mathcal{K}(\mathbf{x}_l, \mathbf{x}_i)}{|P_k|} + \frac{\sum_{r \in P_k} \sum_{s \in P_k} \mathcal{K}(\mathbf{x}_r, \mathbf{x}_s)}{|P_k|^2} \quad (2.12)$$

Com isso, a função objetivo J , dada pela equação (2.10), pode ser reescrita como sendo:

$$J^{(t)} = \sum_{k=1}^K \sum_{i \in P_k^{(t)}} \left\{ \mathcal{K}(\mathbf{x}_i, \mathbf{x}_i) - 2 \frac{\sum_{l \in P_k^{(t)}} \mathcal{K}(\mathbf{x}_l, \mathbf{x}_i)}{|P_k^{(t)}|} + \frac{\sum_{r \in P_k^{(t)}} \sum_{s \in P_k^{(t)}} \mathcal{K}(\mathbf{x}_r, \mathbf{x}_s)}{|P_k^{(t)}|^2} \right\} \quad (2.13)$$

Como não podemos calcular os centroides dos grupos no espaço de características, a etapa de atualização dos protótipos no algoritmo não existe. A etapa de atualização da melhor partição \mathbf{P} se dá através do mapeamento implícito realizado pela equação (2.12). A partição $\mathbf{P} = \{P_1, \dots, P_K\}$ que minimiza a função objetivo J , dada pela equação (2.13), é atualizada de acordo com a seguinte regra:

$$P_k^{(t)} = \{i \in \Omega : \|\Phi(\mathbf{x}_i) - (\mathbf{v}_k^\Phi)^{(t)}\|^2 \leq \|\Phi(\mathbf{x}_i) - (\mathbf{v}_h^\Phi)^{(t)}\|^2, \forall h \neq k, h = 1, \dots, K\} \quad (2.14)$$

O algoritmo a seguir resume o passo a passo iterativo para obter o agrupamento pelo método *kernel* K -médias no espaço de características.

Algoritmo 3: MÉTODO *kernel* K -MÉDIAS NO ESPAÇO DE CARACTERÍSTICAS

(1) Inicialização

Fixe K (o número de grupos), $2 \leq K < n$; Escolha aleatoriamente uma partição \mathbf{P} de Ω em K grupos P_1, P_2, \dots, P_K ; Calcule a matriz *kernel* \mathbf{K} baseado em alguma função *kernel*;

(2) Atualização da partição

$test \leftarrow 0$

para $i = 1$ até n faça

defina o grupo vencedor P_h tal que

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

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).

Métodos de agrupamento para dados simbólicos do tipo intervalo

3.1 Introdução

Com o crescente desenvolvimento nas tecnologias de armazenamento, bem como na velocidade e capacidade dos sistemas, o surgimento de enormes conjunto de dados se tornou corriqueiro e nesse caso a análise de dados clássicos, muitas das vezes, não tem sido apropriada para a extração de conhecimentos úteis, isso por que o esforço computacional necessário para manipular essas bases de dados ainda é um problema, apesar do alto poder de processamento dos computadores atuais (BILLARD; DIDAY, 2003).

Uma abordagem capaz de lidar com o problema anterior seria a redução do conjunto de dados em um conjunto menor de tamanho gerenciável, de modo que este conjunto resumo mantenha o máximo de informações possíveis que estejam contidas nos dados originais. Contudo, ao fazer essa redução, os dados que até então tinham representação clássica passarão a constituir intervalos, lista e afins, sendo exemplo do que chamamos de dados simbólicos, objeto de estudo da Análise de Dados Simbólicos (ADS).

Os primeiros trabalhos referente a abordagem de dados simbólicos surgiram por volta dos anos 80, na qual podemos destacar o trabalho de Diday (1986), e desde então diversos trabalhos começaram surgir. A Análise de dados Simbólicos (ADS) foi desenvolvida para

lidar com enormes conjuntos de dados, no sentido de prover técnicas para redução dessas bases e posteriori análises e extração de informações. Pode-se dizer que a ADS surgiu sob a influência conjunta de três campos: Análise Exploratória de Dados (DIDAY, 1982; WARWICK; MORINEAU, 1984), Inteligência Artificial (MICHALSKI, 1973; RUSSELL; NORVIG, 2016) e Taxonomia Numérica (SNEATH; SOKAL et al., 1973; HAYES-ROTH; MCDERMOTT, 1978). Bock e Diday (2000) apresentam os principais conceitos de ADS e os principais métodos estatísticos desenvolvidos para dados desse tipo.

Diferentemente dos dados clássicos, o uso de dados simbólicos permite atribuir múltiplas regras e valores para as variáveis de modo que o conhecimento extraído seja o máximo possível. COSTA (2011, p.24) afirma que “Essas novas variáveis (conjuntos, intervalos e histogramas) tornam possível reter informações sobre a variabilidade intrínseca ou incerteza do conjunto de dados original”. O princípio básico para a construção de dados simbólicos é que o máximo possível de informações seja mantido e, simultaneamente, que a nova base de dados seja gerenciável. O resultado final é uma nova tabela na qual denominamos tabela de dados simbólicos, que por sua vez apresenta uma estrutura mais complexa que os dados originais, não no sentido de dimensionalidade, mas por causa que essas tabelas podem conter intervalos, conjuntos, funções probabilísticas e etc. As colunas dessas tabelas são chamadas de variáveis simbólicas que descrevem os objetos representados pelas linhas da tabela, que por sua vez podem conter informações a respeito de indivíduos ou algo mais complexo como descrever um conjunto de indivíduos.

Tabela 3.1: Tabela de dados clássicos

i	Sexo	Estado Civil	Idade	Peso	Time
1	M	S	23	80,7	SPO
2	M	C	29	75,5	PAL
3	F	S	36	65,0	FLA
4	F	S	25	62,4	FLA
5	M	C	60	85,9	COR
6	F	C	40	59,7	SPO
7	M	S	45	95,5	SPO
8	M	C	22	76,3	COR
9	F	C	37	62,0	FLA
10	M	C	75	56,8	PAL
11	M	S	15	77,1	COR
12	F	S	19	61,7	FLA

A tabela 3.1 de dados clássicos trata de um exemplo fictício que será utilizado para o entendimento de dados simbólicos, na qual será extraído um conjunto de dados simbólicos a partir dessa base. Cabe ressaltar que este caso é apenas a nível explicativo, de modo que em bases pequenas como esta, os métodos estatísticos clássicos podem ser empregados adequadamente. Considere como exemplo o caso de descrever o peso para “homens casa-

dos”. Baseado na tabela 3.1 o resultado obtido é a lista $\{56, 8, 75, 5, 76, 3, 85, 9\}$, em que esses valores poderiam ser vistos como realizações ξ no intervalo $[56, 8; 85, 9]$. Essa última forma de descrever o peso para “homens casados” é uma representação de dados simbólicos. Como as variáveis Sexo e Estado civil contém duas categorias cada, ao considerar a iteração Sexo \times Estado Civil teremos 4 possíveis categorias $(y)|y_k \in C = \{y_1, y_2, y_3, y_4\}$ em que cada uma dessas 4 categorias descrevem os indivíduos que satisfazem a descrição da categoria. A tabela 3.2 apresenta um conjunto de dados simbólicos referente a descrição das categorias y_k , sendo esta uma tabela de dados simbólicos. Observe que a forma que a tabela foi criada permite estruturar os dados de modo a registrar a variação dos valores.

Tabela 3.2: Tabela de dados simbólicos referente a descrição de y_k

$y_k \in C$	Sexo \times Estado Civil	n_k	Idade	Peso	Time
y_1	Fem Solteiro	3	[19; 36]	[61, 7; 65, 0]	{FLA}
y_2	Fem Casado	2	[37; 40]	[59, 7; 62, 0]	{FLA,SPO}
y_3	Masc Solteiro	3	[15; 45]	[77, 1; 95, 5]	{COR,SPO}
y_4	Masc Casado	4	[22; 75]	[56, 8; 85, 9]	{COR,PAL}

3.2 Tipos de variáveis simbólicas

Diferentemente dos dados clássicos, na qual as variáveis assume apenas um valor, as variáveis simbólicas podem assumir: intervalos, conjunto de categorias, histogramas, funções probabilísticas, etc. As variáveis simbólicas dividem-se basicamente em quatro tipos: variáveis multi-valoradas ordinais e não-ordinais, variáveis modais e variáveis do tipo intervalo.

Variáveis simbólicas multi-valoradas não-ordinais

Seja $E = \{w_1, \dots, w_n\}$ um conjunto de n objetos a serem agrupados. Uma variável X é dita ser uma variável multi-valorada não ordinal se para todo $k \in E$, os valores $X(k)$ correspondem a subconjuntos finitos $D : |X(k)| < \infty$. Por exemplo, seja X as instituições públicas que oferecem ensino superior, existentes em k cidades paraibanas, onde $D = \{\text{UFPB}, \text{UFCEG}, \text{IFPB}\}$. Logo, para $k = \text{João Pessoa}$, $X(\text{João Pessoa}) = \{\text{UFPB}, \text{IFPB}\}$, já para $k = \text{Sousa}$, $X(\text{Sousa}) = \{\text{UFCEG}, \text{IFPB}\}$.

Variáveis simbólicas multi-valoradas ordinais

Uma variável X é dita ser multi-valorada ordinal se o conjunto D estabelece uma relação de ordem \prec , de modo que para quaisquer elementos $a, b \in D$, existe a relação $a \prec b$. Comumente usam um variável qualitativa com domínio finito $D = \{a, b, \dots, h\}$, onde $a \prec b \prec \dots \prec h$. $a \prec b$ é interpretado como a antecede b ou a é menor que b , o que nos permite comparar dois elementos pertencentes a D . Por exemplo, $X = \{\text{Escolaridade}\}$ e $D = \{\text{Ensino fundamental, Ensino Médio, Graduação, Pós-Graduação}\}$.

Variáveis simbólicas modais

Seja $E = \{w_1, \dots, w_n\}$ um conjunto de n objetos a serem agrupados. Uma variável X é dita ser uma variável simbólica modal se para todo $k \in E$, os valores $X(k)$ correspondem a subconjuntos finitos $D : |X(k)| < \infty$, além de apresentar para cada categoria $x \in X(k)$ uma frequência, probabilidade ou peso, que indica o quão frequente, provável, ou relevante a categoria x é para o objeto k . Por exemplo, seja X as instituições públicas que oferecem ensino superior, existentes em k cidades paraibanas, onde $D = \{\text{UFPB, UFCG, IFPB}\}$. Logo, para uma cidade t , $X(t) = \{\text{UFPB}(0, 6), \text{IFPB}(0, 4)\}$.

Variáveis simbólicas do tipo intervalo (intervalares)

Uma variável X é dita ser do tipo intervalo se representa uma realização $\xi = [a : b]$, satisfazendo $a \leq b$ e $\{a, b\} \in \mathbb{R}$. As variáveis Idade e Peso da tabela 3.2 são exemplos de variáveis simbólicas do tipo intervalo. Variáveis desse tipo podem ser usadas quando não é possível obter uma medida exata das observações (instrumentos de precisão) ou medidas estimadas por intervalos de confiança.

3.3 Funções de *kernel* para dados simbólicos do tipo intervalo

Considere um conjunto de n objetos, $\Omega = \{1, \dots, n\}$, descritos por p variáveis simbólicas do tipo intervalo, de modo que cada objeto i é representado por um vetor de intervalos $\mathbf{x}_i = ([a_{i1}, b_{i1}], \dots, [a_{ip}, b_{ip}])^\top$. Seja Φ um mapeamento não linear do espaço de entrada X , para um espaço de dimensão maior \mathcal{F} , denominado espaço de características. A escolha

da função *kernel* é essencial para que o problema da não linearidade no espaço original seja combatido no espaço de características. Uma escolha comum é a utilização do *kernel gaussiano* (ou função RBF), que para dados intervalares é definido de duas formas:

(1) Função RBF de uma componente:

$$\mathcal{K}(\mathbf{x}_i, \mathbf{x}_k) = \exp\left(\frac{-\|\mathbf{x}_i - \mathbf{x}_k\|^2}{2\sigma^2}\right), \quad \sigma > 0, \quad (3.1)$$

onde: $\|\mathbf{x}_i - \mathbf{x}_k\|^2 = \sum_{j=1}^p [(a_{ij} - a_{kj})^2 + (b_{ij} - b_{kj})^2]$, na qual $\mathbf{x}_i = ([a_{i1}, b_{i1}], \dots, [a_{ip}, b_{ip}])$ e $\mathbf{x}_k = ([a_{k1}, b_{k1}], \dots, [a_{kp}, b_{kp}])$ são vetores $2p$ -dimensionais que descrevem o i -ésimo e k -ésimo objetos de Ω , respectivamente (COSTA, 2011).

(2) Função RBF de duas componentes:

$$\mathcal{K}(\mathbf{x}_i, \mathbf{x}_k) = \exp\left(\frac{-\|\mathbf{x}_{iI} - \mathbf{x}_{kI}\|^2}{2\sigma^2}\right) + \exp\left(\frac{-\|\mathbf{x}_{iS} - \mathbf{x}_{kS}\|^2}{2\sigma^2}\right), \quad \sigma > 0, \quad (3.2)$$

onde: $\|\mathbf{x}_{iI} - \mathbf{x}_{kI}\|^2 = \sum_{j=1}^p (a_{ij} - a_{kj})^2$ e $\|\mathbf{x}_{iS} - \mathbf{x}_{kS}\|^2 = \sum_{j=1}^p (b_{ij} - b_{kj})^2$, na qual $\mathbf{x}_{iI} = (a_{i1}, \dots, a_{ip})$ e $\mathbf{x}_{iS} = (b_{i1}, \dots, b_{ip})$ são vetores p -dimensionais relacionados aos limites inferiores e superiores, respectivamente, dos intervalos que descrevem o i -ésimo elemento de Ω , e $\mathbf{x}_{kI} = (a_{k1}, \dots, a_{kp})$ e $\mathbf{x}_{kS} = (b_{k1}, \dots, b_{kp})$ são vetores p -dimensionais relacionados aos limites inferiores e superiores, respectivamente, dos intervalos que descrevem o k -ésimo elemento de Ω .

COSTA (2011, p.38) demonstra que a função RBF de duas componentes definida segundo a equação 3.2 também é uma função *kernel*.

3.4 Métodos de agrupamentos para dados simbólicos do tipo intervalo

Nesta seção apresentamos o método K -médias para intervalo e suas versões kernelizadas, considerando as abordagens da kernelização da métrica e do espaço de características.

3.4.1 Método K -médias para intervalos

Seja $\Omega = \{1, \dots, n\}$ um conjunto de n objetos descritos por p variáveis simbólicas do tipo intervalo, de modo que cada objeto \mathbf{x}_i ($i = 1, \dots, n$) é representado por um vetor de intervalos, ou seja, $\mathbf{x}_i = ([a_{i1}, b_{i1}], \dots, [a_{ip}, b_{ip}])^\top$. Semelhantemente, o protótipo do grupo k ($k = 1, \dots, K$) é dado por $\mathbf{v}_k = ([\alpha_{k1}, \beta_{k1}], \dots, [\alpha_{kp}, \beta_{kp}])^\top$. Seja $\mathbf{P} = \{P_1, \dots, P_K\}$ uma partição de Ω em K grupos disjuntos. O método K -médias para intervalos busca encontrar a partição \mathbf{P} que minimiza a seguinte função objetivo:

$$J^{(t)} = \sum_{k=1}^K \sum_{i \in P_k^{(t)}} d^2(\mathbf{x}_i, \mathbf{v}_k^{(t)}), \quad (3.3)$$

$$\text{onde } d^2(\mathbf{x}_i, \mathbf{v}_k^{(t)}) = \sum_{j=1}^p \left[\left(a_{ij} - \alpha_{kj}^{(t)} \right)^2 + \left(b_{ij} - \beta_{kj}^{(t)} \right)^2 \right]$$

É fácil mostrar que as componentes α_{kj} e β_{kj} dos protótipos dos grupos \mathbf{v}_k ($k = 1, \dots, K$) que minimizam o critério J , dado pela equação 3.3, são calculadas das seguintes formas:

$$\alpha_{kj}^{(t+1)} = \frac{1}{n_k} \sum_{i \in P_k^{(t)}} a_{ij} \quad (3.4)$$

$$\beta_{kj}^{(t+1)} = \frac{1}{n_k} \sum_{i \in P_k^{(t)}} b_{ij}, \quad (3.5)$$

onde n_k é o número de observações pertencentes ao grupo k .

O algoritmo a seguir resume o passo a passo iterativo para obter o agrupamento pelo método K -médias para dados simbólicos do tipo intervalo.

Algoritmo 4: MÉTODO K -MÉDIAS PARA INTERVALOS

(1) Inicialização

Fixe K (o número de grupos), $2 \leq K < n$; Escolha K padrões distintos $\mathbf{v}_1, \dots, \mathbf{v}_K$ pertencentes a Ω como protótipos iniciais e aloque cada padrão i de acordo com o protótipo mais próximo \mathbf{v}_h ($h = \arg \min_{1 \leq k \leq K} \sum_{j=1}^p [(a_{ij} - \alpha_{kj})^2 + (b_{ij} - \beta_{kj})^2]$) para obter a partição inicial $\mathbf{P} = \{P_1, \dots, P_K\}$;

(2) Atualização dos protótipos

Atualize as componentes α_{kj} e β_{kj} dos protótipos dos grupos \mathbf{v}_k ($k = 1, \dots, K$) segundo as equações (3.4) e (3.5) ;

(3) Atualização da partição

$test \leftarrow 0$

para $i = 1$ até n faça

defina o grupo vencedor P_h tal que

$$h = \arg \min_{1 \leq k \leq K} d^2(\mathbf{x}_{ij}, \mathbf{v}_{kj})$$

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).

3.4.2 *Kernel* K -médias para dados do tipo intervalo baseado na kernelização da métrica

Seja $\Omega = \{1, \dots, n\}$ um conjunto de n objetos descritos por p variáveis simbólicas do tipo intervalo, de modo que cada objeto \mathbf{x}_i ($i = 1, \dots, n$) é representado por um vetor de intervalos, ou seja, $\mathbf{x}_i = ([a_{i1}, b_{i1}], \dots, [a_{ip}, b_{ip}])^\top$. Semelhantemente, o protótipo do grupo k ($k = 1, \dots, K$) é dado por $\mathbf{v}_k = ([\alpha_{k1}, \beta_{k1}], \dots, [\alpha_{kp}, \beta_{kp}])^\top$. Seja $\mathbf{P} = \{P_1, \dots, P_K\}$ uma partição de Ω em K grupos disjuntos. O método *kernel* K -médias para intervalos busca

encontrar a partição \mathbf{P} que minimiza a seguinte função objetivo:

$$\begin{aligned} J^{(t)} &= \sum_{k=1}^K \sum_{i \in P_k^{(t)}} \|\Phi(\mathbf{x}_i) - \Phi(\mathbf{v}_k^{(t)})\|^2 \\ &= \sum_{k=1}^K \sum_{i \in P_k^{(t)}} \mathcal{K}(\mathbf{x}_i, \mathbf{x}_i) - 2\mathcal{K}(\mathbf{x}_i, \mathbf{v}_k^{(t)}) + \mathcal{K}(\mathbf{v}_k^{(t)}, \mathbf{v}_k^{(t)}) \end{aligned} \quad (3.6)$$

Para a obtenção dos centroides dos grupos é necessário a escolha de uma função *kernel*. Ao escolher o *kernel* gaussiano de uma componente (amplamente mais utilizado), temos que as componentes α_{kj} e β_{kj} dos protótipos dos grupos \mathbf{v}_k ($k = 1, \dots, K$) que minimizam o critério J são atualizadas das seguintes formas:

$$\alpha_{kj}^{(t+1)} = \frac{\sum_{i \in P_k^{(t)}} \mathcal{K}(\mathbf{x}_i, \mathbf{v}_k^{(t)}) a_{ij}}{\sum_{i \in P_k^{(t)}} \mathcal{K}(\mathbf{x}_i, \mathbf{v}_k^{(t)})} \quad (3.7)$$

$$\beta_{kj}^{(t+1)} = \frac{\sum_{i \in P_k^{(t)}} \mathcal{K}(\mathbf{x}_i, \mathbf{v}_k^{(t)}) b_{ij}}{\sum_{i \in P_k^{(t)}} \mathcal{K}(\mathbf{x}_i, \mathbf{v}_k^{(t)})}, \quad (3.8)$$

onde t e $t + 1$ estão relacionados com a t -ésima iteração do algoritmo e sua iteração subsequente.

Após a obtenção dos protótipos \mathbf{v}_k ($k = 1, \dots, K$) o problema agora é encontrar a partição $\mathbf{P} = \{P_1, \dots, P_K\}$ que minimiza J . Como os protótipos estão fixos, a partição $\mathbf{P} = \{P_1, \dots, P_K\}$ que minimiza a função objetivo J , dada pela equação (3.6), é atualizada de acordo com a seguinte regra:

$$P_k^{(t)} = \left\{ i \in \Omega : \|\Phi(\mathbf{x}_i) - \Phi(\mathbf{v}_k^{(t)})\|^2 \leq \|\Phi(\mathbf{x}_i) - \Phi(\mathbf{v}_h^{(t)})\|^2, \forall h \neq k, h = 1, \dots, K \right\} \quad (3.9)$$

O algoritmo a seguir resume o passo a passo iterativo para obter o agrupamento pelo método *kernel* K -médias para dados do tipo intervalo baseado na kernelização da métrica.

Algoritmo 5: *Kernel* K -MÉDIAS PARA DADOS DO TIPO INTERVALO BASEADO NA KERNELIZAÇÃO DA MÉTRICA

(1) Inicialização

Fixe K (o número de grupos), $2 \leq K < n$; Escolha aleatoriamente uma partição \mathbf{P} de Ω em K grupos P_1, P_2, \dots, P_K ou, alternativamente, escolha K padrões distintos $\mathbf{v}_1, \dots, \mathbf{v}_K$ pertencentes a Ω como protótipos iniciais e aloque cada padrão i de acordo com o protótipo mais próximo \mathbf{v}_h ($h = \arg \min_{1 \leq k \leq K} \|\Phi(\mathbf{x}_i) - \Phi(\mathbf{v}_k)\|^2$) para obter a partição inicial $\mathbf{P} = \{P_1, \dots, P_K\}$;

(2) Atualização dos protótipos

Atualize as componentes α_{kj} e β_{kj} dos protótipos dos grupos \mathbf{v}_k ($k = 1, \dots, K$) segundo as equações (3.7) e (3.8) ;

(3) Atualização da partição

$test \leftarrow 0$

para $i = 1$ até n faça

defina o grupo vencedor P_h tal que

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

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).

3.4.3 *Kernel* K -médias para dados do tipo intervalo no espaço de características

Analogamente ao algoritmo *kernel* K -médias para dados do tipo intervalo baseado na kernelização da métrica, este algoritmo busca obter K grupos, de modo a minimizar a função objetivo J dada pela seguinte equação:

$$J^{(t)} = \sum_{k=1}^K \sum_{i \in P_k^{(t)}} \|\Phi(\mathbf{x}_i) - (\mathbf{v}_k^\Phi)^{(t)}\|^2 \quad (3.10)$$

É necessário encontrar os centroides \mathbf{v}_k^Φ ($k = 1, \dots, K$) que minimizam a função objetivo J , dada pela equação (3.10). É fácil mostrar que $\mathbf{v}_k^\Phi \forall k = 1, \dots, K$ é dado por :

$$(\mathbf{v}_k^\Phi)^{(t+1)} = \frac{1}{|P_k^{(t)}|} \sum_{i \in P_k^{(t)}} \Phi(\mathbf{x}_i) \quad (3.11)$$

Como a transformação não-linear Φ não é conhecida de forma explícita, os protótipos no espaço de características \mathbf{v}_k ($k = 1, \dots, K$) não podem ser obtidos através da equação (3.11). No entanto, a distância $\|\Phi(\mathbf{x}_i) - \mathbf{v}_k^\Phi\|^2$ no espaço de características pode ser obtida, utilizando funções *kernel* no espaço original dos padrões, da seguinte forma (FERREIRA, 2013):

$$\|\Phi(\mathbf{x}_i) - \mathbf{v}_k^\Phi\|^2 = \mathcal{K}(\mathbf{x}_i, \mathbf{x}_i) - 2 \frac{\sum_{l \in P_k} \mathcal{K}(\mathbf{x}_l, \mathbf{x}_i)}{|P_k|} + \frac{\sum_{r \in P_k} \sum_{s \in P_k} \mathcal{K}(\mathbf{x}_r, \mathbf{x}_s)}{|P_k|^2} \quad (3.12)$$

Com isso, a função objetivo J , dada pela equação (3.10), pode ser reescrita como sendo:

$$J^{(t)} = \sum_{k=1}^K \sum_{i \in P_k^{(t)}} \left\{ \mathcal{K}(\mathbf{x}_i, \mathbf{x}_i) - 2 \frac{\sum_{l \in P_k^{(t)}} \mathcal{K}(\mathbf{x}_l, \mathbf{x}_i)}{|P_k^{(t)}|} + \frac{\sum_{r \in P_k^{(t)}} \sum_{s \in P_k^{(t)}} \mathcal{K}(\mathbf{x}_r, \mathbf{x}_s)}{|P_k^{(t)}|^2} \right\} \quad (3.13)$$

Como não podemos calcular os centroides dos grupos no espaço de características, a etapa de atualização dos protótipos no algoritmo não existe. A etapa de atualização da melhor partição \mathbf{P} se dá através do mapeamento implícito realizado pela equação (3.12). A partição $\mathbf{P} = \{P_1, \dots, P_K\}$ que minimiza a função objetivo J , dada pela equação (3.13), é atualizada de acordo com a seguinte regra:

$$P_k^{(t)} = \{i \in \Omega : \|\Phi(\mathbf{x}_i) - (\mathbf{v}_k^\Phi)^{(t)}\|^2 \leq \|\Phi(\mathbf{x}_i) - (\mathbf{v}_h^\Phi)^{(t)}\|^2, \forall h \neq k, h = 1, \dots, K\} \quad (3.14)$$

O algoritmo a seguir resume o passo a passo iterativo para obter o agrupamento pelo

método *kernel* K -médias para dados do tipo intervalo no espaço de características.

Algoritmo 6: *Kernel* K -MÉDIAS PARA DADOS DO TIPO INTERVALO NO ESPAÇO DE CARACTERÍSTICAS

(1) Inicialização

Fixe K (o número de grupos), $2 \leq K < n$; Escolha aleatoriamente uma partição \mathbf{P} de Ω em K grupos P_1, P_2, \dots, P_K ; Calcule a matriz *kernel* \mathbf{K} baseado em alguma função *kernel*.

(2) Atualização da partição

$test \leftarrow 0$

para $i = 1$ até n faça

defina o grupo vencedor P_h tal que

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

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).

Agrupamento *hard* baseado em *kernel* com ponderação automática das variáveis para dados intervalares

4.1 Introdução

Com o desenvolvimento da Análise de Dados Simbólicos (ADS) diversas técnicas tem surgido para se adaptar a essa nova forma de representação dos dados. Quando se trata de algoritmos de agrupamento, apesar dos diversos métodos já existentes, ainda existe uma lacuna a ser preenchida. Os métodos tradicionais, seja ele o K -médias ou suas versões kernelizadas, realizam o processo de agrupamento considerando o mesmo peso para todas as variáveis, ou seja, tais métodos consideram que todas as variáveis tem a mesma importância para a construção dos grupos. Não obstante, se tornou comum a existência de conjuntos de dados de alta dimensão, e dessa forma, podem existir variáveis que apresentem pesos pequenos, ou até mesmo desprezíveis, para o processo de agrupamento. Além disso, das variáveis que foram relevantes, podem existir variáveis com maior peso que as demais. Outro ponto chave é que para a construção de cada grupo, pode haver diferentes conjuntos de variáveis importantes, no sentido de que cada variável pode apresentar pesos diferentes para construção de cada grupo. Nesse sentido ao considerar que existe a possibilidade de haver diferenças nos pesos das variáveis para o processo de agrupamento, de modo que seja possível mensurar esses valores, então a performance desses algoritmos

podem ser melhorada.

Esse capítulo trás a principal contribuição deste trabalho. Aqui apresentamos métodos de agrupamento *hard* para dados simbólicos do tipo intervalo baseado em funções *kernel* com ponderação automática das variáveis. Através de distâncias adaptativas que mudam a cada iteração foi possível mensurar a relevância de cada variável para o processo de agrupamento. Essas distâncias foram obtidas como somas quadradas de distâncias euclidianas entre as observações e os centroides de cada grupo. Para ambas as abordagens, kernelização da métrica e agrupamento no espaço de características, foram propostas distâncias adaptativas considerando parametrização por um único vetor de pesos (distâncias adaptativas globais), onde o peso de cada variável era igual para todos os grupos, e parametrização por um vetor de pesos para cada grupo (distância adaptativas locais), onde os pesos eram diferentes para cada grupo. Para a obtenção dos pesos foi considerada dois tipos de restrições: a primeira, que o produto dos pesos das variáveis era igual a um, e a segunda que a soma dos pesos da variáveis era igual a um.

Segundo Schölkopf et al. (2002) se \mathcal{K}_1 e \mathcal{K}_2 são funções *kernel* então $\mathcal{K}_1 + \mathcal{K}_2$ também é um *kernel*, ou seja, a soma de funções *kernel* também é uma função *kernel*. Esse fato permite tratar conjuntos de variáveis diferentemente, de modo a mensurar a relevância de cada uma delas para os grupos. De forma mais clara, considere um objeto descrito por p variáveis e representado por um vetor p -dimensional. De acordo com a constatação feita por Schölkopf et al. (2002) podemos particionar esse vetor em p partes e considerar p diferentes funções *kernel*. Dessa forma, ao considerar uma transformação não linear $\Phi : X \rightarrow \mathcal{F}$, que toma valores no espaço original dos dados X e mapeia em um espaço de mais alta dimensão \mathcal{F} , de acordo com a proposta feita nesse capítulo é plausível escrever $\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) = \sum_{j=1}^p \mathcal{K}_j(\mathbf{x}_{ij}, \mathbf{x}_{kj})$, onde $\mathcal{K} : X \times X \rightarrow \mathbb{R}$ e $\mathcal{K}_j : X_j \times X_j \rightarrow \mathbb{R}$ são funções *kernel* e X e X_j são o espaço original dos dados e o espaço da j -ésima variável, respectivamente. Sendo assim, a distância entre um padrão \mathbf{x}_i e um protótipo \mathbf{v}_k , calculada através de função *kernel*, para a j -ésima variável, pode ser expressa como:

$$\|\phi_j(\mathbf{x}_{ij}) - \phi_j(\mathbf{v}_{kj})\|^2 = \mathcal{K}_j(\mathbf{x}_{ij}, \mathbf{x}_{ij}) - 2\mathcal{K}_j(\mathbf{x}_{ij}, \mathbf{v}_{kj}) + \mathcal{K}_j(\mathbf{v}_{kj}, \mathbf{v}_{kj}), \quad (4.1)$$

onde cada ϕ_j , $j = 1, \dots, p$, representa um mapeamento não linear para um espaço \mathcal{F}_j , referente a j -ésima variável.

Perceba agora que essa nova forma de calcular distâncias (equação 4.1) permite atribuir pesos as variáveis de modo a diferenciar a relevância de cada uma para o processo de agrupamento. Nesse sentido, considere as seguintes medidas de distâncias entre um objeto \mathbf{x}_i e o centróide do k -ésimo grupo \mathbf{v}_k baseadas na kernelização da métrica:

i) Distância não-adaptativa:

$$d^2(\mathbf{x}_i, \mathbf{v}_k) = \sum_{j=1}^p \|\phi_j(\mathbf{x}_{ij}) - \phi_j(\mathbf{v}_{kj})\|^2 \quad (4.2)$$

ii) Distância adaptativa local sob a restrição de que a soma dos pesos das variáveis em cada grupo deve ser igual a um:

$$d_{\lambda_k}^2(\mathbf{x}_i, \mathbf{v}_k) = \sum_{j=1}^p (\lambda_{kj})^\theta \|\phi_j(\mathbf{x}_{ij}) - \phi_j(\mathbf{v}_{kj})\|^2, \quad (4.3)$$

onde $\lambda_k = (\lambda_{k1}, \dots, \lambda_{kp})$ é o vetor de pesos relativo ao k -ésimo grupo, sujeito a:

$$\begin{cases} \lambda_{kj} \in [0, 1] \quad \forall k, j \\ \sum_{j=1}^p \lambda_{kj} = 1 \end{cases}$$

e $\theta \in (1, \infty)$ é um parâmetro que controla o grau de influencia das variáveis para cada grupo, de modo que quanto maior o valor de θ mais próximos serão os pesos das variáveis para o k -ésimo grupo e quando $\theta \rightarrow 1$ os pesos terão influência máxima.

iii) Distância adaptativa global sob a restrição de que a soma dos pesos das variáveis deve ser igual a um:

$$d_{\lambda}^2(\mathbf{x}_i, \mathbf{v}_k) = \sum_{j=1}^p (\lambda_j)^\theta \|\phi_j(\mathbf{x}_{ij}) - \phi_j(\mathbf{v}_{kj})\|^2, \quad (4.4)$$

onde $\lambda = (\lambda_1, \dots, \lambda_p)$ é o vetor de pesos, sujeito a:

$$\begin{cases} \lambda_j \in [0, 1] \quad \forall j \\ \sum_{j=1}^p \lambda_j = 1 \end{cases}$$

e θ é definido como anteriormente.

iv) Distância adaptativa local sob a restrição de que o produto dos pesos das variáveis em cada grupo deve ser igual a um:

$$d_{\lambda_k}^2(\mathbf{x}_i, \mathbf{v}_k) = \sum_{j=1}^p (\lambda_{kj}) \|\phi_j(\mathbf{x}_{ij}) - \phi_j(\mathbf{v}_{kj})\|^2, \quad (4.5)$$

onde $\lambda_k = (\lambda_{k1}, \dots, \lambda_{kp})$ é o vetor de pesos relativo ao k -ésimo grupo, sujeito a:

$$\begin{cases} \lambda_{kj} > 0 \quad \forall k, j \\ \prod_{j=1}^p \lambda_{kj} = 1 \end{cases}$$

- v) Distância adaptativa global sob a restrição de que o produto dos pesos das variáveis deve ser igual a um:

$$d_{\lambda}^2(\mathbf{x}_i, \mathbf{v}_k) = \sum_{j=1}^p (\lambda_j) \|\phi_j(\mathbf{x}_{ij}) - \phi_j(\mathbf{v}_{kj})\|^2, \quad (4.6)$$

onde $\lambda = (\lambda_1, \dots, \lambda_p)$ é o vetor de pesos, sujeito a:

$$\begin{cases} \lambda_j > 0 \quad \forall j \\ \prod_{j=1}^p \lambda_j = 1 \end{cases}$$

Analogamente, na abordagem no espaço de característica é assumido que os protótipos $\mathbf{v}_k^{\Phi} = (\mathbf{v}_{k1}^{\phi_1}, \dots, \mathbf{v}_{kp}^{\phi_p})$, $k = 1, \dots, K$, são obtidos individualmente para cada variável no espaço de características, permitindo estender as considerações feitas anteriormente na abordagem da kernelização da métrica. Sendo assim, considere as seguintes medidas de distâncias entre um padrão \mathbf{x}_i e o centroide do k -ésimo grupo no espaço de características, \mathbf{v}_k^{Φ} :

- vi) Distância não-adaptativa:

$$d^2(\mathbf{x}_i, \mathbf{v}_k^{\Phi}) = \sum_{j=1}^p \|\phi_j(\mathbf{x}_{ij}) - \mathbf{v}_{kj}^{\phi_j}\|^2 \quad (4.7)$$

- vii) Distância adaptativa local sob a restrição de que a soma dos pesos das variáveis em cada grupo deve ser igual a um:

$$d_{\lambda_k}^2(\mathbf{x}_i, \mathbf{v}_k^{\Phi}) = \sum_{j=1}^p (\lambda_{kj})^{\theta} \|\phi_j(\mathbf{x}_{ij}) - \mathbf{v}_{kj}^{\phi_j}\|^2, \quad (4.8)$$

onde $\lambda_k = (\lambda_{k1}, \dots, \lambda_{kp})$ é o vetor de pesos relativo ao k -ésimo grupo, sujeito a:

$$\begin{cases} \lambda_{kj} \in [0, 1] \quad \forall k, j \\ \sum_{j=1}^p \lambda_{kj} = 1 \end{cases}$$

e $\theta \in (1, \infty)$ é um parâmetro que controla o grau de influencia das variáveis para cada grupo, de modo que quanto maior o valor de θ mais próximos serão os pesos das variáveis para o k -ésimo grupo e quando $\theta \rightarrow 1$ os pesos terão influência máxima.

- viii) Distância adaptativa global sob a restrição de que a soma dos pesos das variáveis deve ser igual a um:

$$d_{\lambda}^2(\mathbf{x}_i, \mathbf{v}_k^{\Phi}) = \sum_{j=1}^p (\lambda_j)^{\theta} \|\phi_j(\mathbf{x}_{ij}) - \mathbf{v}_{kj}^{\phi_j}\|^2, \quad (4.9)$$

onde $\lambda = (\lambda_1, \dots, \lambda_p)$ é o vetor de pesos, sujeito a:

$$\begin{cases} \lambda_j \in [0, 1] \quad \forall j \\ \sum_{j=1}^p \lambda_j = 1 \end{cases}$$

e θ é definido como anteriormente.

- ix) Distância adaptativa local sob a restrição de que o produto dos pesos das variáveis em cada grupo deve ser igual a um:

$$d_{\lambda_k}^2(\mathbf{x}_i, \mathbf{v}_k^{\Phi}) = \sum_{j=1}^p (\lambda_{kj}) \|\phi_j(\mathbf{x}_{ij}) - \mathbf{v}_{kj}^{\phi_j}\|^2, \quad (4.10)$$

onde $\lambda_k = (\lambda_{k1}, \dots, \lambda_{kp})$ é o vetor de pesos relativo ao k -ésimo grupo, sujeito a:

$$\begin{cases} \lambda_{kj} > 0 \quad \forall k, j \\ \prod_{j=1}^p \lambda_{kj} = 1 \end{cases}$$

- x) Distância adaptativa global sob a restrição de que o produto dos pesos das variáveis deve ser igual a um:

$$d_{\lambda}^2(\mathbf{x}_i, \mathbf{v}_k^{\Phi}) = \sum_{j=1}^p (\lambda_j) \|\phi_j(\mathbf{x}_{ij}) - \mathbf{v}_{kj}^{\phi_j}\|^2, \quad (4.11)$$

onde $\lambda = (\lambda_1, \dots, \lambda_p)$ é o vetor de pesos, sujeito a:

$$\begin{cases} \lambda_j > 0 \quad \forall j \\ \prod_{j=1}^p \lambda_j = 1 \end{cases}$$

É importante notar que as distâncias definidas pelas equações (4.3) e (4.5) na kernelização da métrica e pelas equações (4.8) e (4.10) no espaço de características, referentes a distância adaptativa local, possibilita que as variáveis tenham pesos diferentes para cada grupo, fornecendo possíveis conjuntos distintos de variáveis importantes para os grupos. Por outro lado, as distâncias definidas pelas equações (4.4) e (4.6) na kernelização da métrica e pelas equações (4.9) e (4.11) no espaço de características, referentes a distância adaptativa global, assume que os pesos das variáveis são iguais para todos os grupos, no sentido de que o conjunto de variáveis importantes é o mesmo para eles.

Para as distâncias que foram definidas sob a restrição de que a soma dos pesos das variáveis deve ser igual a um, existe a presença do parâmetro θ nas equações, o que não é visto para o caso das distâncias que se baseiam na restrição de que o produto dos pesos das variáveis deve ser igual a um. No processo de derivação para encontrar os pesos ótimos, que será apresentado mais a frente, quando não é considerado o parâmetro θ nos casos sob a restrição de que a soma dos pesos das variáveis deve ser igual a um, a componente λ_{kj} ($k = 1, \dots, K$ e $j = 1, \dots, p$), relacionada aos pesos das variáveis, desaparece, impossibilitando encontrar o pesos que minimizam o critério de adequação entre as observações e os protótipos. Dessa forma, o uso do θ nas medidas de distâncias se justifica como uma estratégia algébrica para contornar esse problema.

4.2 Agrupamento *hard* baseado na kernelização da métrica com ponderação automática das variáveis para dados do tipo intervalo

Nesta parte do trabalho é apresentado os métodos de agrupamento *hard* baseado na kernelização da métrica com ponderação automática das variáveis para dados do tipo intervalo. Todos os algoritmos propostos a seguir minimizam uma função objetivo J que mede a adequação entre os padrões e os centroides dos grupos através de uma medida de distância adequada. De forma geral, os métodos propostos a seguir buscam minimizar uma função objetivo dada por:

$$J^{(t)} = \sum_{k=1}^K \sum_{i \in P_k^{(t)}} \varphi^2(\mathbf{x}_i, \mathbf{v}_k^{(t)}) \quad (4.12)$$

, onde $\varphi^2(\cdot)$ é uma das distâncias definidas anteriormente para a abordagem na kernelização da métrica (distâncias definidas pelas equações (4.2),(4.3),(4.4),(4.5) e (4.6)). De

antemão, para os métodos de agrupamento a seguir foi considerado as duas versões do *kernel* gaussiano. Sendo assim, a intuito de simplificação, vale destacar o seguinte :

1. Se considerarmos o *kernel* gaussiano de uma componente, dado pela equação (3.1), temos que $\mathcal{K}(\mathbf{x}_i, \mathbf{x}_i) = 1$, e dessa forma a distância entre um padrão \mathbf{x}_i e um protótipo \mathbf{v}_k , com relação a j -ésima variável, dada pela equação (4.1), pode ser expressa como:

$$\|\phi_j(\mathbf{x}_{ij}) - \phi_j(\mathbf{v}_{kj})\|^2 = 2(1 - \mathcal{K}_j(\mathbf{x}_{ij}, \mathbf{v}_{kj})) \quad (4.13)$$

2. Se considerarmos o *kernel* gaussiano de duas componentes, dado pela equação (3.2), temos que $\mathcal{K}(\mathbf{x}_i, \mathbf{x}_i) = 2$, e dessa forma a distância entre um padrão \mathbf{x}_i e um protótipo \mathbf{v}_k , com relação a j -ésima variável, dada pela equação (4.1), pode ser expressa como:

$$\|\phi_j(\mathbf{x}_{ij}) - \phi_j(\mathbf{v}_{kj})\|^2 = 2(2 - \mathcal{K}_j(\mathbf{x}_{ij}, \mathbf{v}_{kj})) \quad (4.14)$$

De acordo com a escolha de uma medida de distância e um tipo de *kernel* temos diferentes métodos de agrupamento *hard* para dados simbólicos do tipo intervalo na abordagem da kernelização da métrica. Considerando as distâncias dadas pelas equações (4.2),(4.3),(4.4),(4.5) e (4.6), e considerando as funções RBF de uma e duas componentes, temos os seguintes métodos:

- (a) *Kernel* K -médias baseado na distância não-adaptativa kernelizada, dada pela Eq. (4.2), com o uso da função RBF de uma componente (iKKMEE1):

$$J^{(t)} = 2 \sum_{k=1}^K \sum_{i \in P_k^{(t)}} \sum_{j=1}^p \left(1 - \mathcal{K}_j(\mathbf{x}_{ij}, \mathbf{v}_{kj}^{(t)})\right) \quad (4.15)$$

- (b) *Kernel* K -médias baseado na distância adaptativa local kernelizada sob a restrição de que a soma dos pesos das variáveis em cada grupo deve ser igual um, dada pela Eq. (4.3), com o uso da função RBF de uma componente (iKKMEE-LS1):

$$J^{(t)} = 2 \sum_{k=1}^K \sum_{i \in P_k^{(t)}} \sum_{j=1}^p \left(\lambda_{kj}^{(t)}\right)^\theta \left(1 - \mathcal{K}_j(\mathbf{x}_{ij}, \mathbf{v}_{kj}^{(t)})\right) \quad (4.16)$$

- (c) *Kernel* K -médias baseado na distância adaptativa global kernelizada sob a restrição de que a soma dos pesos das variáveis deve ser igual um, dada pela Eq. (4.4), com o uso da função RBF de uma componente (iKKMEE-GS1):

$$J^{(t)} = 2 \sum_{k=1}^K \sum_{i \in P_k^{(t)}} \sum_{j=1}^p \left(\lambda_j^{(t)} \right)^\theta \left(1 - \mathcal{K}_j(\mathbf{x}_{ij}, \mathbf{v}_{kj}^{(t)}) \right) \quad (4.17)$$

- (d) *Kernel K-médias* baseado na distância adaptativa local kernelizada sob a restrição de que o produto dos pesos das variáveis em cada grupo deve ser igual um, dada pela Eq. (4.5), com o uso da função RBF de uma componente (iKKMEE-LP1):

$$J^{(t)} = 2 \sum_{k=1}^K \sum_{i \in P_k^{(t)}} \sum_{j=1}^p \left(\lambda_{kj}^{(t)} \right) \left(1 - \mathcal{K}_j(\mathbf{x}_{ij}, \mathbf{v}_{kj}^{(t)}) \right) \quad (4.18)$$

- (e) *Kernel K-médias* baseado na distância adaptativa global kernelizada sob a restrição de que o produto dos pesos das variáveis deve ser igual um, dada pela Eq. (4.6), com o uso da função RBF de uma componente (iKKMEE-GP1):

$$J^{(t)} = 2 \sum_{k=1}^K \sum_{i \in P_k^{(t)}} \sum_{j=1}^p \left(\lambda_j^{(t)} \right) \left(1 - \mathcal{K}_j(\mathbf{x}_{ij}, \mathbf{v}_{kj}^{(t)}) \right) \quad (4.19)$$

- (f) *Kernel K-médias* baseado na distância não-adaptativa kernelizada, dada pela Eq. (4.2), com o uso da função RBF de duas componentes (iKKMEE2):

$$J^{(t)} = 2 \sum_{k=1}^K \sum_{i \in P_k^{(t)}} \sum_{j=1}^p \left(2 - \mathcal{K}_j(\mathbf{x}_{ij}, \mathbf{v}_{kj}^{(t)}) \right) \quad (4.20)$$

- (g) *Kernel K-médias* baseado na distância adaptativa local kernelizada sob a restrição de que a soma dos pesos das variáveis em cada grupo deve ser igual um, dada pela Eq. (4.3), com o uso da função RBF de duas componentes (iKKMEE-LS2):

$$J^{(t)} = 2 \sum_{k=1}^K \sum_{i \in P_k^{(t)}} \sum_{j=1}^p \left(\lambda_{kj}^{(t)} \right)^\theta \left(2 - \mathcal{K}_j(\mathbf{x}_{ij}, \mathbf{v}_{kj}^{(t)}) \right) \quad (4.21)$$

- (h) *Kernel K-médias* baseado na distância adaptativa global kernelizada sob a restrição de que a soma dos pesos das variáveis deve ser igual um, dada pela Eq. (4.4), com o uso da função RBF de duas componentes (iKKMEE-GS2):

$$J^{(t)} = 2 \sum_{k=1}^K \sum_{i \in P_k^{(t)}} \sum_{j=1}^p \left(\lambda_j^{(t)} \right)^\theta \left(2 - \mathcal{K}_j(\mathbf{x}_{ij}, \mathbf{v}_{kj}^{(t)}) \right) \quad (4.22)$$

- (i) *Kernel K-médias* baseado na distância adaptativa local kernelizada sob a restrição de que o produto dos pesos das variáveis em cada grupo deve ser igual um, dada pela Eq. (4.5), com o uso da função RBF de duas componentes (iKKMEE-LP2):

$$J^{(t)} = 2 \sum_{k=1}^K \sum_{i \in P_k^{(t)}} \sum_{j=1}^p \left(\lambda_{kj}^{(t)} \right) \left(2 - \mathcal{K}_j(\mathbf{x}_{ij}, \mathbf{v}_{kj}^{(t)}) \right) \quad (4.23)$$

- (j) *Kernel K-médias* baseado na distância adaptativa global kernelizada sob a restrição de que o produto dos pesos das variáveis deve ser igual um, dada pela Eq. (4.6), com o uso da função RBF de duas componentes (iKKMEE-GP2):

$$J^{(t)} = 2 \sum_{k=1}^K \sum_{i \in P_k^{(t)}} \sum_{j=1}^p \left(\lambda_j^{(t)} \right) \left(2 - \mathcal{K}_j(\mathbf{x}_{ij}, \mathbf{v}_{kj}^{(t)}) \right) \quad (4.24)$$

Como todas as funções objetivos estão definidas agora é preciso encontrar os valores dos protótipos \mathbf{v}_k , $k = (1, \dots, K)$, que minimizam o critério J . Para tanto é necessário a escolha de uma função de *kernel* e ao optar pelas funções *kernel* de uma e duas componentes obtem-se resultados diferentes. Nessa etapa a partição \mathbf{P} de Ω em K grupos e os pesos das variáveis são mantidos fixos.

Proposição 4.1. *Os componentes α_{kj} e β_{kj} do protótipo \mathbf{v}_{kj} ($k = 1, \dots, K$ e $j = 1, \dots, p$) que minimizam o critério J baseado nas distâncias adaptativa kernelizadas, dadas pelas equações (4.2), (4.3), (4.4), (4.5) e (4.6), são calculados das seguintes formas:*

- i) *Se o kernel gaussiano utilizado for o de uma componente, α_{kj} e β_{kj} são dados por :*

$$\alpha_{kj}^{(t+1)} = \frac{\sum_{i \in P_k^{(t)}} \mathcal{K}_j(\mathbf{x}_{ij}, \mathbf{v}_{kj}^{(t)}) a_{ij}}{\sum_{i \in P_k^{(t)}} \mathcal{K}_j(\mathbf{x}_{ij}, \mathbf{v}_{kj}^{(t)})} \quad (4.25)$$

$$\beta_{kj}^{(t+1)} = \frac{\sum_{i \in P_k^{(t)}} \mathcal{K}_j(\mathbf{x}_{ij}, \mathbf{v}_{kj}^{(t)}) b_{ij}}{\sum_{i \in P_k^{(t)}} \mathcal{K}_j(\mathbf{x}_{ij}, \mathbf{v}_{kj}^{(t)})} \quad (4.26)$$

- ii) *Se o kernel gaussiano utilizado for o de duas componentes, α_{kj} e β_{kj} são dados por*

:

$$\alpha_{kj}^{(t+1)} = \frac{\sum_{i \in P_k^{(t)}} \exp \left(- \sum_{j=1}^p \left(a_{ij} - \alpha_{kj}^{(t)} \right)^2 / 2\sigma^2 \right) a_{ij}}{\sum_{i \in P_k^{(t)}} \exp \left(- \sum_{j=1}^p \left(a_{ij} - \alpha_{kj}^{(t)} \right)^2 / 2\sigma^2 \right)} \quad (4.27)$$

$$\beta_{kj}^{(t+1)} = \frac{\sum_{i \in P_k^{(t)}} \exp \left(- \sum_{j=1}^p \left(b_{ij} - \beta_{kj}^{(t)} \right)^2 / 2\sigma^2 \right) b_{ij}}{\sum_{i \in P_k^{(t)}} \exp \left(- \sum_{j=1}^p \left(b_{ij} - \beta_{kj}^{(t)} \right)^2 / 2\sigma^2 \right)} \quad (4.28)$$

Demonstração. Iremos demonstrar a obtenção da componente α_{kj} quando se usa o *kernel* gaussiano de uma componente, de modo que a obtenção da componente β_{kj} para este mesmo caso e das componentes α_{kj} e β_{kj} para o caso do *kernel* de duas componentes segue o mesmo raciocínio.

O objetivo é minimizar J em relação a α_{kj} . Como o critério J é aditivo devemos encontrar o valor de α_{kj} que minimiza a parcela $2(1 - \mathcal{K}_j(\mathbf{x}_{ij}, \mathbf{v}_{kj}))$, e isso equivale a maximizar $\mathcal{K}_j(\mathbf{x}_{ij}, \mathbf{v}_{kj}) \forall i \in P_k$, pois $\mathcal{K}_j(\mathbf{x}_{ij}, \mathbf{v}_{kj}) \in (0, 1]$. Assim, para o k -ésimo grupo devemos encontrar o componente α_{kj} que maximiza a parcela:

$$\sum_{i \in P_k} \mathcal{K}_j(\mathbf{x}_{ij}, \mathbf{v}_{kj}) \quad (4.29)$$

Derivando a parcela dada em (4.29) em relação a α_{kj} temos que :

$$\begin{aligned} \frac{\partial \sum_{i \in P_k} \mathcal{K}_j(\mathbf{x}_{ij}, \mathbf{v}_{kj})}{\partial \alpha_{kj}} &= \sum_{i \in P_k} \frac{\partial \left\{ \exp \left(- \sum_{j=1}^p \left[(a_{ij} - \alpha_{kj})^2 + (b_{ij} - \beta_{kj})^2 \right] / 2\sigma^2 \right) \right\}}{\partial \alpha_{kj}} \\ &= - \sum_{i \in P_k} \underbrace{\exp \left(\frac{- \sum_{j=1}^p \left[(a_{ij} - \alpha_{kj})^2 + (b_{ij} - \beta_{kj})^2 \right]}{2\sigma^2} \right)}_{\mathcal{K}_j(\mathbf{x}_{ij}, \mathbf{v}_{kj})} \frac{(a_{ij} - \alpha_{kj})}{\sigma^2} \\ &= \frac{-1}{\sigma^2} \sum_{i \in P_k} \mathcal{K}_j(\mathbf{x}_{ij}, \mathbf{v}_{kj}) (a_{ij} - \alpha_{kj}) \end{aligned}$$

Igualando o resultado encontrado acima a zero, temos que, para $k = 1, \dots, K$ e $j = 1, \dots, p$:

$$\begin{aligned} \frac{-1}{\sigma^2} \sum_{i \in P_k} \mathcal{K}_j(\mathbf{x}_{ij}, \mathbf{v}_{kj}) (a_{ij} - \alpha_{kj}) &= 0 \\ \sum_{i \in P_k} \mathcal{K}_j(\mathbf{x}_{ij}, \mathbf{v}_{kj}) (a_{ij} - \alpha_{kj}) &= 0 \\ \sum_{i \in P_k} \mathcal{K}_j(\mathbf{x}_{ij}, \mathbf{v}_{kj}) a_{ij} - \alpha_{kj} \sum_{i \in P_k} \mathcal{K}_j(\mathbf{x}_{ij}, \mathbf{v}_{kj}) &= 0 \end{aligned}$$

$$\begin{aligned} \sum_{i \in P_k} \mathcal{K}_j(\mathbf{x}_{ij}, \mathbf{v}_{kj}) a_{ij} &= \alpha_{kj} \sum_{i \in P_k} \mathcal{K}_j(\mathbf{x}_{ij}, \mathbf{v}_{kj}) \\ \alpha_{kj} &= \frac{\sum_{i \in P_k} \mathcal{K}_j(\mathbf{x}_{ij}, \mathbf{v}_{kj}) a_{ij}}{\sum_{i \in P_k} \mathcal{K}_j(\mathbf{x}_{ij}, \mathbf{v}_{kj})} \end{aligned}$$

□

Para os métodos baseado em distâncias adaptativas o problema agora é determinar os pesos das variáveis que minimizam a função objetivo J , sob as restrições adequadas. Nessa parte do algoritmo, tanto a partição \mathbf{P} quanto os centroides \mathbf{v}_k , $k = 1, \dots, K$, são mantidos fixos.

Proposição 4.2. *Os pesos das variáveis que minimizam a função objetivo geral J , dada pela equação (4.12), dependem da distância adaptativa utilizada:*

(i) *Se a distância adaptativa kernelizada for dada pela equação (4.3), gerando os métodos iKKMEE-LS1 e iKKMEE-LS2, os pesos $\boldsymbol{\lambda}_k = (\lambda_{k1}, \dots, \lambda_{kp})$ que minimizam o critério J , dado pela equação (4.16) ou pela equação (4.21), sob as restrições $\lambda_{kj} \in [0, 1] \forall k, j$ e $\sum_{j=1}^p \lambda_{kj} = 1 \forall k$, tem seus componentes λ_{kj} ($k = 1, \dots, K$ e $j = 1, \dots, p$) calculados da seguinte forma:*

$$\lambda_{kj}^{(t+1)} = \left[\sum_{l=1}^p \left(\frac{\sum_{i \in P_k^{(t)}} \|\phi_j(\mathbf{x}_{ij}) - \phi_j(\mathbf{v}_{kj}^{(t+1)})\|^2}{\sum_{i \in P_k^{(t)}} \|\phi_l(\mathbf{x}_{il}) - \phi_l(\mathbf{v}_{kl}^{(t+1)})\|^2} \right)^{\frac{1}{\theta-1}} \right]^{-1} \quad (4.30)$$

(ii) *Se a distância adaptativa kernelizada for dada pela equação (4.4), gerando os métodos iKKMEE-GS1 e iKKMEE-GS2, os pesos $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_p)$ que minimizam*

o critério J , dado pela equação (4.17) ou pela equação (4.22), sob as restrições $\lambda_j \in [0, 1] \forall j$ e $\sum_{j=1}^p \lambda_{kj} = 1$, tem seus componentes λ_j ($j = 1, \dots, p$) calculados da seguinte forma:

$$\lambda_j^{(t+1)} = \left[\sum_{l=1}^p \left(\frac{\sum_{k=1}^K \sum_{i \in P_k^{(t)}} \|\phi_j(\mathbf{x}_{ij}) - \phi_j(\mathbf{v}_{kj}^{(t+1)})\|^2}{\sum_{k=1}^K \sum_{i \in P_k^{(t)}} \|\phi_l(\mathbf{x}_{il}) - \phi_l(\mathbf{v}_{kl}^{(t+1)})\|^2} \right)^{\frac{1}{\theta-1}} \right]^{-1} \quad (4.31)$$

(iii) Se a distância adaptativa kernelizada for dada pela equação (4.5), gerando os métodos *iKKMEE-LP1* e *iKKMEE-LP2*, os pesos $\boldsymbol{\lambda}_k = (\lambda_{k1}, \dots, \lambda_{kp})$ que minimizam o critério J , dado pela equação (4.18) ou pela equação (4.23), sob as restrições $\lambda_{kj} > 0 \forall k, j$ e $\prod_{j=1}^p \lambda_{kj} = 1 \forall k$, tem seus componentes λ_{kj} ($k = 1, \dots, K$ e $j = 1, \dots, p$) calculados da seguinte forma:

$$\lambda_{kj}^{(t+1)} = \frac{\left\{ \prod_{l=1}^p \left(\sum_{i \in P_k^{(t)}} \|\phi_l(\mathbf{x}_{il}) - \phi_l(\mathbf{v}_{kl}^{(t+1)})\|^2 \right) \right\}^{\frac{1}{p}}}{\sum_{i \in P_k^{(t)}} \|\phi_j(\mathbf{x}_{ij}) - \phi_j(\mathbf{v}_{kj}^{(t+1)})\|^2} \quad (4.32)$$

(iv) Se a distância adaptativa kernelizada for dada pela equação (4.6), gerando os métodos *iKKMEE-GP1* e *iKKMEE-GP2*, os pesos $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_p)$ que minimizam o critério J , dado pela equação (4.19) ou pela equação (4.24), sob as restrições $\lambda_j > 0 \forall j$ e $\prod_{j=1}^p \lambda_{kj} = 1$, tem seus componentes λ_j ($j = 1, \dots, p$) calculados da seguinte forma:

$$\lambda_j^{(t+1)} = \frac{\left\{ \prod_{l=1}^p \left(\sum_{k=1}^K \sum_{i \in P_k^{(t)}} \|\phi_l(\mathbf{x}_{il}) - \phi_l(\mathbf{v}_{kl}^{(t+1)})\|^2 \right) \right\}^{\frac{1}{p}}}{\sum_{k=1}^K \sum_{i \in P_k^{(t)}} \|\phi_j(\mathbf{x}_{ij}) - \phi_j(\mathbf{v}_{kj}^{(t+1)})\|^2} \quad (4.33)$$

Demonstração.

(i) Queremos minimizar

$$J = 2 \sum_{k=1}^K \sum_{i \in P_k} \sum_{j=1}^p (\lambda_{kj})^\theta \|\phi_j(\mathbf{x}_{ij}) - \phi_j(\mathbf{v}_{kj})\|^2$$

com relação a λ_{kj} ($k = 1, 2, \dots, K$ e $j = 1, 2, \dots, p$) sob a restrição de que $\lambda_{kj} \in [0, 1] \forall k, j$ e $\sum_{j=1}^p \lambda_{kj} = 1 \forall k$. A partição \mathbf{P} de Ω em K grupos está fixa, bem como os protótipos \mathbf{v}_k , $k = 1, \dots, K$, e o parâmetro θ . Dessa forma, podemos rescrever J da seguinte forma:

$$J = \sum_{k=1}^K F_k(\boldsymbol{\lambda}_k) = \sum_{k=1}^K \sum_{j=1}^p (\lambda_{kj})^\theta \sum_{i \in P_k} \|\phi_j(\mathbf{x}_{ij}) - \phi_j(\mathbf{v}_{kj})\|^2,$$

onde: $F_k(\boldsymbol{\lambda}_k) = F_k(\lambda_{k1}, \dots, \lambda_{kp}) = \sum_{j=1}^p (\lambda_{kj})^\theta M_{kj}$, $M_{kj} = \sum_{i \in P_k} \|\phi_j(\mathbf{x}_{ij}) - \phi_j(\mathbf{v}_{kj})\|^2$

Como o critério J é aditivo, o problema se torna minimizar $F_k = \sum_{j=1}^p (\lambda_{kj})^\theta M_{kj}$, $k = 1, \dots, K$. Considere $G_k(\lambda_{k1}, \dots, \lambda_{kp}) = \sum_{j=1}^p \lambda_{kj} - 1$. Queremos encontrar os extremos de J sob a restrição $G_k(\lambda_{k1}, \dots, \lambda_{kp}) = 0$. Utilizando os conceitos do método de multiplicadores de Lagrange, temos:

$$\nabla F_k(\lambda_{k1}, \dots, \lambda_{kp}) = \mu \nabla G_k(\lambda_{k1}, \dots, \lambda_{kp})$$

Então, para $k = 1, \dots, K$ e $j = 1, \dots, p$, temos:

$$\begin{aligned} \frac{\partial F_k(\lambda_{k1}, \dots, \lambda_{kp})}{\partial \lambda_{kj}} &= \mu \frac{\partial G_k(\lambda_{k1}, \dots, \lambda_{kp})}{\partial \lambda_{kj}} \\ \theta (\lambda_{kj})^{\theta-1} M_{kj} &= \mu \\ \lambda_{kj} &= \left(\frac{\mu}{\theta}\right)^{\frac{1}{\theta-1}} \cdot \frac{1}{(M_{kj})^{\frac{1}{\theta-1}}} \end{aligned} \quad (4.34)$$

Usando a restrição de que, $\sum_{l=1}^p \lambda_{kl} = 1$, $\forall k$, temos:

$$\begin{aligned} \sum_{l=1}^p \left(\frac{\mu}{\theta}\right)^{\frac{1}{\theta-1}} \cdot \frac{1}{(M_{kl})^{\frac{1}{\theta-1}}} &= 1 \\ \left(\frac{\mu}{\theta}\right)^{\frac{1}{\theta-1}} &= \left[\sum_{l=1}^p \frac{1}{(M_{kl})^{\frac{1}{\theta-1}}} \right]^{-1} \end{aligned} \quad (4.35)$$

Substituindo (4.35) em (4.34) temos :

$$\begin{aligned}
 \lambda_{kj} &= \left[\sum_{l=1}^p \frac{1}{(M_{kl})^{\frac{1}{\theta-1}}} \right]^{-1} \left[(M_{kj})^{\frac{1}{\theta-1}} \right]^{-1} \\
 &= \left[\sum_{l=1}^p \frac{(M_{kj})^{\frac{1}{\theta-1}}}{(M_{kl})^{\frac{1}{\theta-1}}} \right]^{-1} \\
 &= \left[\sum_{l=1}^p \left(\frac{M_{kj}}{M_{kl}} \right)^{\frac{1}{\theta-1}} \right]^{-1} \\
 \\ \\
 \lambda_{kj} &= \left[\sum_{l=1}^p \left(\frac{\sum_{i \in P_k} \|\phi_j(\mathbf{x}_{ij}) - \phi_j(\mathbf{v}_{kj})\|^2}{\sum_{i \in P_k} \|\phi_l(\mathbf{x}_{il}) - \phi_l(\mathbf{v}_{kl})\|^2} \right)^{\frac{1}{\theta-1}} \right]^{-1}
 \end{aligned}$$

Veja que $\frac{\partial F_k}{\partial \lambda_{kj}} = \theta(\lambda_{kj})^{\theta-1} M_{kj}$, então $\frac{\partial^2 F_k}{\partial (\lambda_{kj})^2} = \theta(\theta-1)(\lambda_{kj})^{\theta-2} M_{kj}$ e $\frac{\partial^2 F_k}{\partial (\lambda_{kj}) \partial (\lambda_{kl})} = 0 \quad \forall l \neq j$

A matriz Hessiana de F_k avaliada em $\boldsymbol{\lambda}_k = (\lambda_{k1}, \dots, \lambda_{kp})$ é

$$H(\boldsymbol{\lambda}_k) = \begin{pmatrix} \frac{\theta(\theta-1)M_{k1}}{\sum_{l=1}^p \left(\frac{M_{k1}}{M_{kl}}\right)^{\frac{\theta-2}{\theta-1}}} & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{\theta(\theta-1)M_{kp}}{\sum_{l=1}^p \left(\frac{M_{kp}}{M_{kl}}\right)^{\frac{\theta-2}{\theta-1}}} \end{pmatrix}$$

de modo que a matriz $H(\boldsymbol{\lambda}_k)$ é positiva-definida, levando a conclusão de que esse extremo é ponto de mínimo.

(ii) A demonstração é similar ao da parte (i).

(iii) Queremos minimizar

$$J = 2 \sum_{k=1}^K \sum_{i \in P_k} \sum_{j=1}^p (\lambda_{kj}) \|\phi_j(\mathbf{x}_{ij}) - \phi_j(\mathbf{v}_{kj})\|^2$$

com relação a λ_{kj} ($k = 1, 2, \dots, K$ e $j = 1, 2, \dots, p$) sob a restrição de que $\lambda_{kj} > 0 \quad \forall k, j$ e $\prod_{j=1}^p \lambda_{kj} = 1 \quad \forall k$. A partição \mathbf{P} de Ω em K grupos está fixa, bem como os protótipos \mathbf{v}_k , $k = 1, \dots, K$. Dessa forma, podemos rescrever J da seguinte forma:

$$J = \sum_{k=1}^K F_k(\boldsymbol{\lambda}_k) = \sum_{k=1}^K \sum_{j=1}^p (\lambda_{kj}) \sum_{i \in P_k} \|\phi_j(\mathbf{x}_{ij}) - \phi_j(\mathbf{v}_{kj})\|^2$$

$$, \text{ onde: } F_k(\boldsymbol{\lambda}_k) = F_k(\lambda_{k1}, \dots, \lambda_{kp}) = \sum_{j=1}^p (\lambda_{kj}) M_{kj}, \quad M_{kj} = \sum_{i \in P_k} \|\phi_j(\mathbf{x}_{ij}) - \phi_j(\mathbf{v}_{kj})\|^2$$

Como o critério J é aditivo, o problema se torna minimizar $F_k = \sum_{j=1}^p (\lambda_{kj}) M_{kj}$, $k = 1, \dots, K$. Considere $G_k(\lambda_{k1}, \dots, \lambda_{kp}) = \prod_{j=1}^p \lambda_{kj} - 1$. Queremos encontrar os extremos de J considerando $G_k(\lambda_{k1}, \dots, \lambda_{kp}) = 0$. Ao fazer isso, temos:

$$\begin{aligned} \prod_{j=1}^p \lambda_{kj} &= 1 \\ \log \left(\prod_{j=1}^p \lambda_{kj} \right) &= \log(1) \\ \sum_{j=1}^p \log(\lambda_{kj}) &= 0 \end{aligned}$$

Considere agora $H_k(\lambda_{k1}, \dots, \lambda_{kp}) = \sum_{j=1}^p \log(\lambda_{kj})$. Queremos encontrar os extremos de J considerando $H_k(\lambda_{k1}, \dots, \lambda_{kp}) = 0$. Utilizando os conceitos do método de multiplicadores de Lagrange, temos:

$$\nabla F_k(\lambda_{k1}, \dots, \lambda_{kp}) = \mu \nabla H_k(\lambda_{k1}, \dots, \lambda_{kp})$$

Então, para $k = 1, \dots, K$ e $j = 1, \dots, p$, temos:

$$\begin{aligned} \frac{\partial F_k(\lambda_{k1}, \dots, \lambda_{kp})}{\partial \lambda_{kj}} &= \mu \frac{\partial g_k(\lambda_{k1}, \dots, \lambda_{kp})}{\partial \lambda_{kj}} \\ M_{kj} &= \frac{\mu}{\lambda_{kj}} \\ \lambda_{kj} &= \frac{\mu}{M_{kj}} \end{aligned} \tag{4.36}$$

Usando a restrição de que $\prod_{l=1}^p \lambda_{kl} = 1$, $\forall k$, temos :

$$\begin{aligned}
 \prod_{l=1}^p \frac{\mu}{M_{kl}} &= 1 \\
 \mu^p \prod_{l=1}^p \left(\frac{1}{M_{kl}} \right) &= 1 \\
 \mu &= \left(\prod_{l=1}^p M_{kl} \right)^{\frac{1}{p}}
 \end{aligned} \tag{4.37}$$

Substituindo (4.37) em (4.36) obtemos :

$$\lambda_{kj} = \frac{\left\{ \prod_{l=1}^p M_{kl} \right\}^{\frac{1}{p}}}{M_{kj}} = \frac{\left\{ \prod_{l=1}^p \left(\sum_{i \in P_k} \|\phi_l(\mathbf{x}_{il}) - \phi_l(\mathbf{v}_{kl})\|^2 \right) \right\}^{\frac{1}{p}}}{\sum_{i \in P_k} \|\phi_j(\mathbf{x}_{ij}) - \phi_j(\mathbf{v}_{kj})\|^2}$$

Observe que $F_k(\lambda_{k1}, \dots, \lambda_{kp}) = \sum_{j=1}^p \lambda_{kj} M_{kj} = p \{M_{k1} \times \dots \times M_{kp}\}^{\frac{1}{p}}$ é o ponto extremo. Como $F_k(1, \dots, 1) = \sum_{j=1}^p M_{kj} = M_{k1} + \dots + M_{kp}$ e sabendo que a média geométrica é menor que a média aritmética, i.e., $\{M_{k1} \times \dots \times M_{kp}\}^{\frac{1}{p}} < \frac{1}{p} \{M_{k1} + \dots + M_{kp}\}$ (a igualdade vale quando $M_{k1} = \dots = M_{kp}$), podemos concluir que o ponto extremo é um mínimo.

(iv) A demonstração é similar a demonstração da parte (iii). □

O próximo passo consiste em alocar os n padrões nos K grupos, de modo que nessa etapa os protótipos \mathbf{v}_k , $k = 1, \dots, K$, e os pesos λ_{kj} , $k = 1, \dots, K$, $j = 1, \dots, p$ são mantidos fixos.

Proposição 4.3. *A partição $\mathbf{P} = (P_1, \dots, P_K)$ independe da medida de distância utilizada. Para qualquer distância baseada na kernelização da métrica os grupos P_k , $k = 1, \dots, K$, que minimizam o critério J , são atualizados da seguinte forma:*

$$P_k^{(t)} = \left\{ i \in \Omega : \varphi^2(\mathbf{x}_i, \mathbf{v}_k^{(t)}) \leq \varphi^2(\mathbf{x}_i, \mathbf{v}_h^{(t)}), \forall h \neq k, h = 1, \dots, K \right\} \tag{4.38}$$

onde, $\varphi^2(\cdot)$ é uma das distâncias definida pelas equações (4.2), (4.3), (4.4), (4.5) ou (4.6).

Demonstração. A demonstração é obtida de forma direta. □

O algoritmo a seguir resume o passo a passo iterativo para obter o agrupamento pelos métodos baseado na kernelização da métrica propostos nessa seção.

Algoritmo 7: ALGORITMO PARA A OBTENÇÃO DO AGRUPAMENTO PELOS MÉTODOS PROPOSTOS NA KERNELIZAÇÃO DA MÉTRICA

(1) Inicialização

Fixe K (o número de grupos), $2 \leq K < n$; Fixe o valor de θ , $1 < \theta < \infty$, (se a restrição considerada for a de que a soma dos pesos das variáveis deve ser igual a um); escolha aleatoriamente uma partição \mathbf{P} de Ω em K grupos P_1, P_2, \dots, P_K , ou alternativamente, escolha k padrões distintos $\mathbf{v}_1, \dots, \mathbf{v}_K$ pertencendo a Ω como protótipos iniciais e aloque cada padrão i de acordo com o protótipo mais próximo $\mathbf{v}_h (h = \arg \min_{1 \leq k \leq K} \varphi^2(\mathbf{x}_i, \mathbf{v}_k))$, onde $\varphi^2(\cdot)$ é uma das distâncias dada pelas equações (4.2),(4.3),(4.4),(4.5) ou (4.6)) para obter a partição inicial $\mathbf{P} = \{P_1, P_2, \dots, P_K\}$; Inicialize todos os pesos igual $\frac{1}{p}$ se a restrição for a de que a soma dos pesos das variáveis é igual a um, ou todos os pesos igual a um, se a restrição for de que o produto do peso das variáveis deve ser igual a um;

(2) Atualização dos protótipos

Atualize os protótipos dos grupos de acordo com as equações (4.25) e (4.26) se a função RBF utilizada for de uma componente, ou segundo as equações (4.27) e (4.28) se a função RBF utilizada for de duas componentes;

(3) Atualização das distâncias

Se o método considerado usar a distancia não-adaptativa (Equação (4.2)) vá para o passo (4). Caso contrário, atualize os pesos das variáveis, de acordo com as equações (4.30),(4.31),(4.32) ou (4.33), dependendo da distância utilizada;

(4) Atualização da partição

$test \leftarrow 0$

para $i = 1$ até n faça

defina o grupo vencedor P_h tal que

$h = \arg \min_{1 \leq k \leq K} \varphi^2(\mathbf{x}_i, \mathbf{v}_k)$, onde $\varphi^2(\cdot)$ é uma medida de distancia dada pelas equações (4.2), (4.3), (4.4), (4.5) ou (4.6).

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).

4.3 Agrupamento *hard* no espaço de características com ponderação automática das variáveis para dados do tipo intervalo

Nesta seção é apresentado os métodos de agrupamento *hard* considerando a abordagem do espaço de características com ponderação automática das variáveis para dados do tipo intervalo. Todos os algoritmos propostos a seguir minimizam uma função objetivo J que mede a adequação entre os padrões e os centroides dos grupos através de uma medida de distância adequada. De forma geral, os métodos propostos a seguir buscam minimizar uma função objetivo dada por:

$$J^{(t)} = \sum_{k=1}^K \sum_{i \in P_k} \varphi^2(\mathbf{x}_i, (\mathbf{v}_k^\Phi)^{(t)}) , \quad (4.39)$$

onde $\varphi^2(\cdot)$ é uma das distâncias definidas anteriormente para a abordagem no espaço de características (distância definidas pelas equações (4.7),(4.8),(4.9),(4.10) e (4.11)).

De acordo com a escolha de uma medida de distância temos diferentes métodos de agrupamento *hard* para dados simbólicos do tipo intervalo na abordagem do espaço de características. Considerando as distâncias dadas pelas equações (4.7),(4.8),(4.9),(4.10) e (4.11), e considerando as funções RBF de uma e duas componentes, temos os seguintes métodos:

- (a) *Kernel K*-médias baseado na distância não-adaptativa no espaço de características, dada pela Eq. (4.7), com o uso da função RBF de uma componente (iKKMEC1):

$$J^{(t)} = \sum_{k=1}^K \sum_{i \in P_k^{(t)}} \sum_{j=1}^p \|\phi_j(\mathbf{x}_{ij}) - (\mathbf{v}_{kj}^{\phi_j})^{(t)}\|^2 \quad (4.40)$$

- (b) *Kernel K*-médias baseado na distância adaptativa local no espaço de características sob a restrição de que a soma dos pesos das variáveis em cada grupo deve ser igual um, dada pela Eq. (4.8), com o uso da função RBF de uma componente (iKKMEC-LS1):

$$J^{(t)} = \sum_{k=1}^K \sum_{i \in P_k^{(t)}} \sum_{j=1}^p \left(\lambda_{kj}^{(t)} \right)^\theta \|\phi_j(\mathbf{x}_{ij}) - (\mathbf{v}_{kj}^{\phi_j})^{(t)}\|^2 \quad (4.41)$$

- (c) *Kernel K*-médias baseado na distância adaptativa global no espaço de características sob a restrição de que a soma dos pesos das variáveis deve ser igual um, dada pela Eq. (4.9), com o uso da função RBF de uma componente (iKKMEC-GS1):

$$J^{(t)} = \sum_{k=1}^K \sum_{i \in P_k^{(t)}} \sum_{j=1}^p \left(\lambda_j^{(t)} \right)^\theta \|\phi_j(\mathbf{x}_{ij}) - (\mathbf{v}_{kj}^{\phi_j})^{(t)}\|^2 \quad (4.42)$$

- (d) *Kernel K*-médias baseado na distância adaptativa local no espaço de características sob a restrição de que o produto dos pesos das variáveis em cada grupo deve ser igual um, dada pela Eq. (4.10), com o uso da função RBF de uma componente (iKKMEC-LP1):

$$J^{(t)} = \sum_{k=1}^K \sum_{i \in P_k^{(t)}} \sum_{j=1}^p \left(\lambda_{kj}^{(t)} \right) \|\phi_j(\mathbf{x}_{ij}) - (\mathbf{v}_{kj}^{\phi_j})^{(t)}\|^2 \quad (4.43)$$

- (e) *Kernel K*-médias baseado na distância adaptativa global no espaço de características sob a restrição de que o produto dos pesos das variáveis deve ser igual um, dada pela Eq. (4.11), com o uso da função RBF de uma componente (iKKMEC-GP1):

$$J^{(t)} = \sum_{k=1}^K \sum_{i \in P_k^{(t)}} \sum_{j=1}^p \left(\lambda_j^{(t)} \right) \|\phi_j(\mathbf{x}_{ij}) - (\mathbf{v}_{kj}^{\phi_j})^{(t)}\|^2 \quad (4.44)$$

- (f) *Kernel K*-médias baseado na distância não-adaptativa no espaço de características, dada pela Eq. (4.7), com o uso da função RBF de duas componentes (iKKMEC2):

$$J^{(t)} = \sum_{k=1}^K \sum_{i \in P_k^{(t)}} \sum_{j=1}^p \|\phi_j(\mathbf{x}_{ij}) - (\mathbf{v}_{kj}^{\phi_j})^{(t)}\|^2 \quad (4.45)$$

- (g) *Kernel K*-médias baseado na distância adaptativa local no espaço de características sob a restrição de que a soma dos pesos das variáveis em cada grupo deve ser igual um, dada pela Eq. (4.8), com o uso da função RBF de duas componentes (iKKMEC-LS2):

$$J^{(t)} = \sum_{k=1}^K \sum_{i \in P_k^{(t)}} \sum_{j=1}^p \left(\lambda_{kj}^{(t)} \right)^\theta \|\phi_j(\mathbf{x}_{ij}) - (\mathbf{v}_{kj}^{\phi_j})^{(t)}\|^2 \quad (4.46)$$

- (h) *Kernel K*-médias baseado na distância adaptativa global no espaço de características sob a restrição de que a soma dos pesos das variáveis deve ser igual um, dada pela Eq. (4.9), com o uso da função RBF de duas componentes (iKKMEC-GS2):

$$J^{(t)} = \sum_{k=1}^K \sum_{i \in P_k^{(t)}} \sum_{j=1}^p \left(\lambda_j^{(t)} \right)^\theta \|\phi_j(\mathbf{x}_{ij}) - (\mathbf{v}_{kj}^{\phi_j})^{(t)}\|^2 \quad (4.47)$$

- (i) *Kernel K-médias* baseado na distância adaptativa local no espaço de características sob a restrição de que o produto dos pesos das variáveis em cada grupo deve ser igual um, dada pela Eq. (4.10), com o uso da função RBF de duas componentes (iKKMEC-LP2):

$$J^{(t)} = \sum_{k=1}^K \sum_{i \in P_k^{(t)}} \sum_{j=1}^p \left(\lambda_{kj}^{(t)} \right) \|\phi_j(\mathbf{x}_{ij}) - (\mathbf{v}_{kj}^{\phi_j})^{(t)}\|^2 \quad (4.48)$$

- (j) *Kernel K-médias* baseado na distância adaptativa global no espaço de características sob a restrição de que o produto dos pesos das variáveis deve ser igual um, dada pela Eq. (4.11), com o uso da função RBF de duas componentes (iKKMEC-GP2):

$$J^{(t)} = \sum_{k=1}^K \sum_{i \in P_k^{(t)}} \sum_{j=1}^p \left(\lambda_j^{(t)} \right) \|\phi_j(\mathbf{x}_{ij}) - (\mathbf{v}_{kj}^{\phi_j})^{(t)}\|^2 \quad (4.49)$$

É preciso encontrar os valores dos protótipos \mathbf{v}_k^Φ , $k = (1, \dots, K)$, que minimizam o critério J . Nesse passo, a partição \mathbf{P} e os pesos das variáveis são mantidos fixos.

Proposição 4.4. *Os componentes $\mathbf{v}_{kj}^{\phi_j}$ ($k = 1, \dots, K$ e $j = 1, \dots, p$) que minimizam o critério J , baseado em qualquer distância no espaço de características (Eqs. (4.7), (4.8), (4.9), (4.10) ou (4.11)), são calculados da seguinte forma:*

$$(\mathbf{v}_{kj}^{\phi_j})^{(t+1)} = \frac{\sum_{i \in P_k^{(t)}} \phi_j(\mathbf{x}_{ij})}{|P_k^{(t)}|} \quad (4.50)$$

Demonstração. Como o critério J é aditivo, o problema se torna encontrar $\mathbf{v}_{kj}^{\phi_j}$ ($k = 1, \dots, K$ e $j = 1, \dots, p$) que minimiza a parcela:

$$\sum_{i \in P_k} \|\phi_j(\mathbf{x}_{ij}) - \mathbf{v}_{kj}^{\phi_j}\|^2 \quad (4.51)$$

Derivando o termo dado em (4.51) com relação a $\mathbf{v}_{kj}^{\phi_j}$ e igualando a zero, temos que, para $k = 1, \dots, K$ e $j = 1, \dots, p$:

$$\begin{aligned} \frac{\partial \sum_{i \in P_k} \|\phi_j(\mathbf{x}_{ij}) - \mathbf{v}_{kj}^{\phi_j}\|^2}{\partial \mathbf{v}_{kj}^{\phi_j}} &= 0 \\ -2 \sum_{i \in P_k} \left(\phi_j(\mathbf{x}_{ij}) - \mathbf{v}_{kj}^{\phi_j} \right) &= 0 \\ \sum_{i \in P_k} \phi_j(\mathbf{x}_{ij}) - \sum_{i \in P_k} \mathbf{v}_{kj}^{\phi_j} &= 0 \\ \mathbf{v}_{kj}^{\phi_j} &= \frac{\sum_{i \in P_k} \phi_j(\mathbf{x}_{ij})}{|P_k|} \end{aligned}$$

onde $|P_k|$ é a cardinalidade do grupo k -ésimo grupo. \square

De modo semelhante aos métodos de agrupamentos convencionais, as transformações não lineares ϕ_1, \dots, ϕ_p não são conhecidas de forma explícita e dessa forma não podemos obter as componentes $\mathbf{v}_{kj}^{\phi_j}$ do protótipo \mathbf{v}_k^Φ , $k = 1, \dots, K$ calculadas pela equação (4.50). No entanto, a informação dessas componentes dos protótipos é implicitamente considerada, de modo que, ao utilizar funções de *kernel*, a distância $\|\phi_j(\mathbf{x}_{ij}) - \mathbf{v}_{kj}^{\phi_j}\|^2$ no espaço de características é calculado no espaço original da seguinte forma:

$$\begin{aligned} \|\phi_j(\mathbf{x}_{ij}) - \mathbf{v}_{kj}^{\phi_j}\|^2 &= \phi_j(\mathbf{x}_{ij}) \cdot \phi_j(\mathbf{x}_{ij}) - 2 \cdot \phi_j(\mathbf{x}_{ij}) \cdot \mathbf{v}_{kj}^{\phi_j} + \mathbf{v}_{kj}^{\phi_j} \cdot \mathbf{v}_{kj}^{\phi_j} \\ &= \phi_j(\mathbf{x}_{ij}) \cdot \phi_j(\mathbf{x}_{ij}) - 2 \cdot \phi_j(\mathbf{x}_{ij}) \cdot \frac{1}{|P_k|} \sum_{l \in P_k} \phi_j(\mathbf{x}_{lj}) \\ &\quad + \frac{1}{|P_k|} \sum_{r \in P_k} \phi_j(\mathbf{x}_{rj}) \cdot \frac{1}{|P_k|} \sum_{s \in P_k} \phi_j(\mathbf{x}_{sj}) \\ &= \phi_j(\mathbf{x}_{ij}) \cdot \phi_j(\mathbf{x}_{ij}) - \frac{2}{|P_k|} \sum_{l \in P_k} \phi_j(\mathbf{x}_{ij}) \cdot \phi_j(\mathbf{x}_{lj}) \\ &\quad + \frac{1}{|P_k|^2} \sum_{r \in P_k} \sum_{s \in P_k} \phi_j(\mathbf{x}_{rj}) \cdot \phi_j(\mathbf{x}_{sj}) \\ &= \mathcal{K}_j(\mathbf{x}_{ij}, \mathbf{x}_{ij}) - \frac{2}{|P_k|} \sum_{l \in P_k} \mathcal{K}_j(\mathbf{x}_{ij}, \mathbf{x}_{lj}) + \frac{1}{|P_k|^2} \sum_{r \in P_k} \sum_{s \in P_k} \mathcal{K}_j(\mathbf{x}_{rj}, \mathbf{x}_{sj}) \end{aligned}$$

Para os métodos baseados em distâncias adaptativas o próximo passo é determinar os pesos das variáveis que minimizam o critério geral J , dado pela equação (4.39), sob as restrições adequadas. Nessa etapa, tanto a partição $\mathbf{P} = \{P_1, \dots, P_K\}$ quanto os protótipos no espaço de características \mathbf{v}_k^Φ , $k = 1, \dots, K$, são mantidos fixos.

Proposição 4.5. *Os pesos das variáveis que minimizam a função objetivo geral J , dada*

pela equação (4.39), dependem da distância adaptativa utilizada no espaço de características:

(i) Se a distância adaptativa no espaço de características for dada pela equação (4.8), gerando os métodos *iKKMEC-LS1* e *iKKMEC-LS2*, os pesos $\boldsymbol{\lambda}_k = (\lambda_{k1}, \dots, \lambda_{kp})$ que minimizam o critério J , dado pela equação (4.41) ou pela equação (4.46), sob as restrições $\lambda_{kj} \in [0, 1] \forall k, j$ e $\sum_{j=1}^p \lambda_{kj} = 1 \forall k$, tem seus componentes λ_{kj} ($k = 1, \dots, K$ e $j = 1, \dots, p$) calculados da seguinte forma:

$$\lambda_{kj}^{(t+1)} = \left[\sum_{l=1}^p \left(\frac{\sum_{i \in P_k^{(t)}} \|\phi_j(\mathbf{x}_{ij}) - (\mathbf{v}_{kj}^{\phi_j})^{(t+1)}\|^2}{\sum_{i \in P_k^{(t)}} \|\phi_l(\mathbf{x}_{il}) - (\mathbf{v}_{kl}^{\phi_l})^{(t+1)}\|^2} \right)^{\frac{1}{\theta-1}} \right]^{-1} \quad (4.52)$$

(ii) Se a distância adaptativa no espaço de características for dada pela equação (4.9), gerando os métodos *iKKMEC-GS1* e *iKKMEC-GS2*, os pesos $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_p)$ que minimizam o critério J , dado pela equação (4.42) ou pela equação (4.47), sob as restrições $\lambda_j \in [0, 1] \forall j$ e $\sum_{j=1}^p \lambda_j = 1$, tem seus componentes λ_j ($j = 1, \dots, p$) calculados da seguinte forma:

$$\lambda_j^{(t+1)} = \left[\sum_{l=1}^p \left(\frac{\sum_{k=1}^K \sum_{i \in P_k^{(t)}} \|\phi_j(\mathbf{x}_{ij}) - (\mathbf{v}_{kj}^{\phi_j})^{(t+1)}\|^2}{\sum_{k=1}^K \sum_{i \in P_k^{(t)}} \|\phi_l(\mathbf{x}_{il}) - (\mathbf{v}_{kl}^{\phi_l})^{(t+1)}\|^2} \right)^{\frac{1}{\theta-1}} \right]^{-1} \quad (4.53)$$

(iii) Se a distância adaptativa no espaço de características for dada pela equação (4.10), gerando os métodos *iKKMEC-LP1* e *iKKMEC-LP2*, os pesos $\boldsymbol{\lambda}_k = (\lambda_{k1}, \dots, \lambda_{kp})$ que minimizam o critério J , dado pela equação (4.43) ou pela equação (4.48), sob as restrições $\lambda_{kj} > 0 \forall k, j$ e $\prod_{j=1}^p \lambda_{kj} = 1 \forall k$, tem seus componentes λ_{kj} ($k = 1, \dots, K$ e $j = 1, \dots, p$) calculados da seguinte forma:

$$\lambda_{kj}^{(t+1)} = \frac{\left\{ \prod_{l=1}^p \left(\sum_{i \in P_k^{(t)}} \|\phi_l(\mathbf{x}_{il}) - (\mathbf{v}_{kl}^{\phi_l})^{(t+1)}\|^2 \right) \right\}^{\frac{1}{p}}}{\sum_{i \in P_k^{(t)}} \|\phi_j(\mathbf{x}_{ij}) - (\mathbf{v}_{kj}^{\phi_j})^{(t+1)}\|^2} \quad (4.54)$$

(iv) Se a distância adaptativa no espaço de características for dada pela equação (4.11), gerando os métodos *iKKMEC-GP1* e *iKKMEC-GP2*, os pesos $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_p)$ que

minimizam o critério J , dado pela equação (4.44) ou pela equação (4.49), sob as restrições $\lambda_j > 0 \forall j$ e $\prod_{j=1}^p \lambda_{kj} = 1$, tem seus componentes λ_j ($j = 1, \dots, p$) calculados da seguinte forma:

$$\lambda_j^{(t+1)} = \frac{\left\{ \prod_{l=1}^p \left(\sum_{k=1}^K \sum_{i \in P_k^{(t)}} \|\phi_l(\mathbf{x}_{il}) - (\mathbf{v}_{kl}^{\phi_l})^{(t+1)}\|^2 \right) \right\}^{\frac{1}{p}}}{\sum_{k=1}^K \sum_{i \in P_k^{(t)}} \|\phi_j(\mathbf{x}_{ij}) - (\mathbf{v}_{kj}^{\phi_j})^{(t+1)}\|^2} \quad (4.55)$$

Demonstração. A demonstração dessa proposição é similar a demonstração da proposição 4.1. \square

O último passo agora é determinar a partição \mathbf{P} de Ω em K grupos que minimiza a função objetivo J , dado pela equação (4.39). Nessa etapa do algoritmo os centroides no espaço de características \mathbf{v}_k^Φ , $k = 1, \dots, K$, e os pesos das variáveis estão fixos.

Proposição 4.6. *A partição $\mathbf{P} = (P_1, \dots, P_K)$ independe da medida de distância utilizada. Para qualquer distância baseada no espaço de características, os grupos P_k , $k = 1, \dots, K$, que minimizam o critério J , são atualizados da seguinte forma:*

$$P_k^{(t)} = \{i \in \Omega : \varphi^2(\mathbf{x}_i, (\mathbf{v}_k^\Phi)^{(t)}) \leq \varphi^2(\mathbf{x}_i, (\mathbf{v}_h^\Phi)^{(t)}), \forall h \neq k, h = 1, \dots, K\} \quad (4.56)$$

onde, $\varphi^2(\cdot)$ é uma das distâncias definida pelas equações (4.7), (4.8), (4.9), (4.10) ou (4.11).

Demonstração. A demonstração é obtida de forma direta. \square

O algoritmo a seguir resume o passo a passo iterativo para obter o agrupamento pelos métodos baseado no espaço de características propostos nessa seção.

Algoritmo 8: ALGORITMO PARA A OBTENÇÃO DO AGRUPAMENTO PELOS MÉTODOS PROPOSTOS NO ESPAÇO DE CARACTERÍSTICAS

(1) Inicialização

Fixe K (o número de grupos), $2 \leq K < n$; Fixe o valor de θ , $1 < \theta < \infty$, (se a restrição considerada for a de que a soma dos pesos das variáveis deve ser igual a um); escolha aleatoriamente uma partição \mathbf{P} de Ω em K grupos P_1, P_2, \dots, P_K ; Inicialize todos os pesos igual $1/p$ se a restrição for de que a soma dos pesos das variáveis é igual a um, ou inicialize todos os pesos igual a um se a restrição for de que o produto do peso das variáveis deve ser igual a um; para cada variável j escolha uma função de *kernel* (nesse trabalho utilizou-se o *kernel* gaussiano com uma e duas componentes) e calcule a matriz *kernel* \mathbf{K}_j , de modo que o elemento na linha i e coluna l dessa matriz é dado por $\mathcal{K}_j(\mathbf{x}_{ij}, \mathbf{x}_{lj}) \forall i, l = 1, \dots, n$;

(2) Atualização das distâncias

Se o método considerado usar a distancia não-adaptativa (4.7) vá para o passo (3). Caso contrário, atualize os pesos das variáveis, de acordo com as equações (4.52),(4.53),(4.54) ou (4.55) dependendo da distância utilizada;

(3) Atualização da partição

$test \leftarrow 0$

para $i = 1$ até n faça

defina o grupo vencedor P_h tal que

$h = \arg \min_{1 \leq k \leq K} \varphi^2(\mathbf{x}_i, \mathbf{v}_k^\Phi)$, onde $\varphi^2(\cdot)$ é uma medida de distancia dada pelas equações (4.7), (4.8), (4.9), (4.10) ou (4.11).

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).

4.4 Convergência dos métodos de agrupamento *hard* baseado em *kernel* com ponderação automática das variáveis para dados do tipo intervalo

Para estudar a convergência dos métodos de agrupamento *hard* propostos nesse capítulo iremos nos restringir ao algoritmo *kernel* K-médias baseado na distância adaptativa local kernelizada sob a restrição de que a soma dos pesos das variáveis em cada grupo deve ser igual um, com o uso da função RBF de uma componente (iKKMEE-LS1), de modo que o que é observado para este algoritmo se estende para os demais. Para este algoritmo a função objetivo é dada por:

$$\begin{aligned} J(\mathbf{V}, \mathbf{\Lambda}, \mathbf{P}) &= \sum_{k=1}^K \sum_{i \in P_k} d_{\lambda_k}^2(\mathbf{x}_i, \mathbf{v}_k) \\ &= 2 \sum_{k=1}^K \sum_{i \in P_k} \sum_{j=1}^p (\lambda_{kj})^\theta (1 - \mathcal{K}_j(\mathbf{x}_{ij}, \mathbf{v}_{kj})) \end{aligned}$$

onde, $\mathbf{V} = (\mathbf{v}_1, \dots, \mathbf{v}_K)$ é uma K -upla de centroides, $\mathbf{\Lambda} = (\boldsymbol{\lambda}_1, \dots, \boldsymbol{\lambda}_K)$ é uma K -upla de pesos e $\mathbf{P} = (P_1, \dots, P_K)$ é uma partição de Ω em K grupos.

Defina $\mathbf{V}^* = (\mathbf{v}_1^*, \dots, \mathbf{v}_K^*)$, $\mathbf{\Lambda}^* = (\boldsymbol{\lambda}_1^*, \dots, \boldsymbol{\lambda}_K^*)$ e $\mathbf{P}^* = (P_1^*, \dots, P_K^*)$ da forma que

$$J(\mathbf{V}^*, \mathbf{\Lambda}^*, \mathbf{P}^*) = \min \{ J(\mathbf{V}, \mathbf{\Lambda}, \mathbf{P}) : \mathbf{V} \in \mathbb{R}^{K \times 2p}, \mathbf{\Lambda} \in [0, 1]^{K \times p}, \mathbf{P} \in \mathbb{P}_K \}$$

onde, \mathbb{P}_K é o conjunto de todas as partições de Ω em K grupos.

Diday e Simon (1976) afirmam que as propriedades de convergência de algoritmos desse tipo podem ser estudadas a partir de duas séries: $y_t = (\mathbf{V}^{(t)}, \mathbf{\Lambda}^{(t)}, \mathbf{P}^{(t)})$ e $z_t = J(y_t) = J(\mathbf{V}^{(t)}, \mathbf{\Lambda}^{(t)}, \mathbf{P}^{(t)})$ ($t = 0, 1, 2, \dots$). A partir de uma parcela inicial $y_0 = (\mathbf{V}^0, \mathbf{\Lambda}^0, \mathbf{P}^0)$, o algoritmo calcula sucessivos termos da série y_t até J convergir para um valor estacionário.

Proposição 4.7. *A série z_t decresce a cada iteração do algoritmo e converge.*

Demonstração. De início iremos demonstrar que a série $z_t = J(y_t)$ decresce a cada iteração do algoritmo. Para tanto, considere as seguintes desigualdades:

$$J(\mathbf{V}^{(t)}, \mathbf{\Lambda}^{(t)}, \mathbf{P}^{(t)}) \stackrel{\text{(I)}}{\geq} J(\mathbf{V}^{(t+1)}, \mathbf{\Lambda}^{(t)}, \mathbf{P}^{(t)}) \stackrel{\text{(II)}}{\geq} J(\mathbf{V}^{(t+1)}, \mathbf{\Lambda}^{(t+1)}, \mathbf{P}^{(t)}) \stackrel{\text{(III)}}{\geq} J(\mathbf{V}^{(t+1)}, \mathbf{\Lambda}^{(t+1)}, \mathbf{P}^{(t+1)})$$

A desigualdade (I) vale porque:

$$J\left(\mathbf{V}^{(t)}, \boldsymbol{\Lambda}^{(t)}, \mathbf{P}^{(t)}\right) = \sum_{k=1}^K \sum_{i \in P_k^{(t)}} d_{\lambda_k^{(t)}}^2\left(\mathbf{x}_i, \mathbf{v}_k^{(t)}\right),$$

$$J\left(\mathbf{V}^{(t+1)}, \boldsymbol{\Lambda}^{(t)}, \mathbf{P}^{(t)}\right) = \sum_{k=1}^K \sum_{i \in P_k^{(t)}} d_{\lambda_k^{(t)}}^2\left(\mathbf{x}_i, \mathbf{v}_k^{(t+1)}\right)$$

e de acordo com a proposição 4.1,

$$\mathbf{V}^{(t+1)} = \left(\mathbf{v}_1^{(t+1)}, \dots, \mathbf{v}_K^{(t+1)}\right) = \underbrace{\arg \min}_{\mathbf{V}=(\mathbf{v}_1, \dots, \mathbf{v}_K)} \sum_{k=1}^K \sum_{i \in P_k^{(t)}} d_{\lambda_k^{(t)}}^2\left(\mathbf{x}_i, \mathbf{v}_k\right).$$

Além disso, a desigualdade (II) também vale porque

$$J\left(\mathbf{V}^{(t+1)}, \boldsymbol{\Lambda}^{(t+1)}, \mathbf{P}^{(t)}\right) = \sum_{k=1}^K \sum_{i \in P_k^{(t)}} d_{\lambda_k^{(t+1)}}^2\left(\mathbf{x}_i, \mathbf{v}_k^{(t+1)}\right),$$

e pela proposição 4.2,

$$\boldsymbol{\Lambda}^{(t+1)} = \left(\lambda_1^{(t+1)}, \dots, \lambda_K^{(t+1)}\right) = \underbrace{\arg \min}_{\boldsymbol{\Lambda}=(\lambda_1, \dots, \lambda_K)} \sum_{k=1}^K \sum_{i \in P_k^{(t)}} d_{\lambda_k}^2\left(\mathbf{x}_i, \mathbf{v}_k^{(t+1)}\right).$$

Por fim, a desigualdade (III) vale porque

$$J\left(\mathbf{V}^{(t+1)}, \boldsymbol{\Lambda}^{(t+1)}, \mathbf{P}^{(t+1)}\right) = \sum_{k=1}^K \sum_{i \in P_k^{(t+1)}} d_{\lambda_k^{(t+1)}}^2\left(\mathbf{x}_i, \mathbf{v}_k^{(t+1)}\right),$$

e de acordo com a proposição 4.3

$$\mathbf{P}^{(t+1)} = \left(P_1^{(t+1)}, \dots, P_K^{(t+1)}\right) = \underbrace{\arg \min}_{\mathbf{P}=(P_1, \dots, P_K)} \sum_{k=1}^K \sum_{i \in P_k} d_{\lambda_k^{(t+1)}}^2\left(\mathbf{x}_i, \mathbf{v}_k^{(t+1)}\right).$$

Como a série decresce e é limitada no zero ($z_t > 0$), ela converge. □

Proposição 4.8. *A série $y_t = \left(\mathbf{V}^{(t)}, \boldsymbol{\Lambda}^{(t)}, \mathbf{P}^{(t)}\right)$ converge.*

Demonstração. Suponha que a estacionaridade de z_t seja atingida na iteração T , de modo que $z_T = z_{T+h}$, para todo h inteiro e $h \geq 1$. Fazendo $h = 1$, temos que $J(y_T) = J(y_{T+1})$, ou seja $J(\mathbf{V}^{(T)}, \mathbf{\Lambda}^{(T)}, \mathbf{P}^{(T)}) = J(\mathbf{V}^{(T+1)}, \mathbf{\Lambda}^{(T+1)}, \mathbf{P}^{(T+1)})$ e segundo a proposição 4.7 podemos rescrever essa igualdade como:

$$\underbrace{J(\mathbf{V}^{(T)}, \mathbf{\Lambda}^{(T)}, \mathbf{P}^{(T)})}_{z_T} \stackrel{(I)}{=} J(\mathbf{V}^{(T+1)}, \mathbf{\Lambda}^{(T)}, \mathbf{P}^{(T)}) \stackrel{(II)}{=} J(\mathbf{V}^{(T+1)}, \mathbf{\Lambda}^{(T+1)}, \mathbf{P}^{(T)}) \stackrel{(III)}{=} \underbrace{J(\mathbf{V}^{(T+1)}, \mathbf{\Lambda}^{(T+1)}, \mathbf{P}^{(T+1)})}_{z_{T+1}}$$

□

Na igualdade (I) $\mathbf{P}^{(T)}$ e $\mathbf{\Lambda}^{(T)}$ estão fixos, de modo que o único minimizando de J é \mathbf{V} , levando a conclusão de que $\mathbf{V}^{(T)} = \mathbf{V}^{(T+1)}$. Da igualdade (II), temos que $\mathbf{\Lambda}^{(T)} = \mathbf{\Lambda}^{(T+1)}$, pois, como $\mathbf{P}^{(T)}$ e $\mathbf{V}^{(T+1)}$ estão fixos, $\mathbf{\Lambda}$ é o único minimizando de J . Por último, da igualdade (III), temos que $\mathbf{P}^{(T)} = \mathbf{P}^{(T+1)}$ pois \mathbf{P} é a única minimizando de J quando $\mathbf{V}^{(T+1)}$ e $\mathbf{\Lambda}^{(T+1)}$ estão fixos (aqui é necessário usar a ordem lexicográfica para a alocação no caso de igualdade de distâncias entre os padrões e centroides). Visto isso, concluímos que $y_T = y_{T+1}$. Estendendo o resultado, podemos concluir que $y_T = y_{T+h}$, para todo $h \geq 1$ inteiro, e dessa forma segue que y_t converge.

5.1 Introdução

Neste capítulo será apresentada uma avaliação experimental dos métodos de agrupamento *hard* baseado em *kernel* com ponderação automática das variáveis para dados intervalares. Para avaliar a performance desses algoritmos em relação aos métodos convencionais foi realizado uma simulação de Monte Carlo com dois conjuntos de dados simulados e foram realizadas aplicações a 12 conjuntos de dados reais intervalares. Todos os experimentos realizados nesse trabalho foram feitos utilizando a linguagem de programação R (R Core Team, 2013).

5.1.1 Conjuntos de dados simulados do tipo intervalo

Para demonstrar a eficiência dos métodos de agrupamento *hard* baseado em *kernel* com ponderação automática das variáveis para dados intervalares foi realizado um experimento de Monte Carlo considerando duas diferentes configurações de conjunto de dados. Em ambos os casos foi considerado que os dados eram descritos por três variáveis simbólicas.

Os conjuntos de dados sintéticos intervalares foram gerados considerando que cada

observação $\mathbf{x}_{ij} = [a_{ij}, b_{ij}]$, é fruto de um intervalo do tipo $[w_{ij} - \gamma_{ij}/2, w_{ij} + \gamma_{ij}/2]$, onde γ_{ij} é gerado a partir de um intervalo predefinido. Nesse trabalho consideramos três intervalos: $[0.1, 0.2]$, $[0.1, 1.5]$ e $[1, 5]$. Os valores w_{ij} foram gerados a partir de distribuições normais que serão detalhadas a seguir. Em ambos conjuntos de dados simulados, os parâmetros foram selecionados de forma que houvessem subespaços importantes e não importantes no processo de agrupamento.

Conjunto de dados simulados 1

O conjunto de dados simulados 1 é composto por 200 observações que estão divididas entre quatro grupos com 50 observações que são descritas por três variáveis simbólicas do tipo intervalo. Os dados foram gerados a partir de uma distribuição normal trivariada, onde para cada grupo foi definido um vetor de médias e uma matriz de covariância específicos. Nesse exemplo foi considerado que as variáveis eram independentes. As informações das médias e variâncias estão dispostas na tabela 5.1.

Tabela 5.1: Informações das estruturas de média e variância do conjunto de dados simulados 1.

$\boldsymbol{\mu}$	Grupo 1	Grupo 2	Grupo 3	Grupo 4
μ_1	-0,5	0,5	0,0	0,0
μ_2	-0,5	-0,5	0,5	0,5
μ_3	0,0	0,0	-0,5	0,5
$\boldsymbol{\Sigma}$	Grupo 1	Grupo 2	Grupo 3	Grupo 4
σ_1^2	0,04	0,04	1,00	1,00
σ_2^2	0,04	0,04	0,04	0,04
σ_3^2	1,00	1,00	0,04	0,04

Nessa configuração cada grupo está bem definido em apenas duas dimensões. Observe através da figura 5.1 que os grupos 1 e 2 são descritos pela variável \mathbf{x}_1 , de modo que esta é relevante para a definição desses grupos (ver figura 5.1 (b)). Por outro lado, os grupos 3 e 4 são descritos pela variável \mathbf{x}_3 , ou seja, esta é relevante para a construção desses grupos (ver figura 5.1 (c)). Ao observar a figura 5.1 (d) percebe-se que a variável \mathbf{x}_3 causa sobreposição dos dados para os grupos 1 e 2, enquanto a variável \mathbf{x}_1 causa sobreposição para os grupos 3 e 4.

Cabe ressaltar que a figura 5.1 foi construída através de um conjunto considerando o intervalo $[0.1, 0.2]$, de modo que quando os valores do intervalo aumenta os dados intervalares tendem a se sobreporem, dificultando a definição dos grupos como comentado anteriormente.

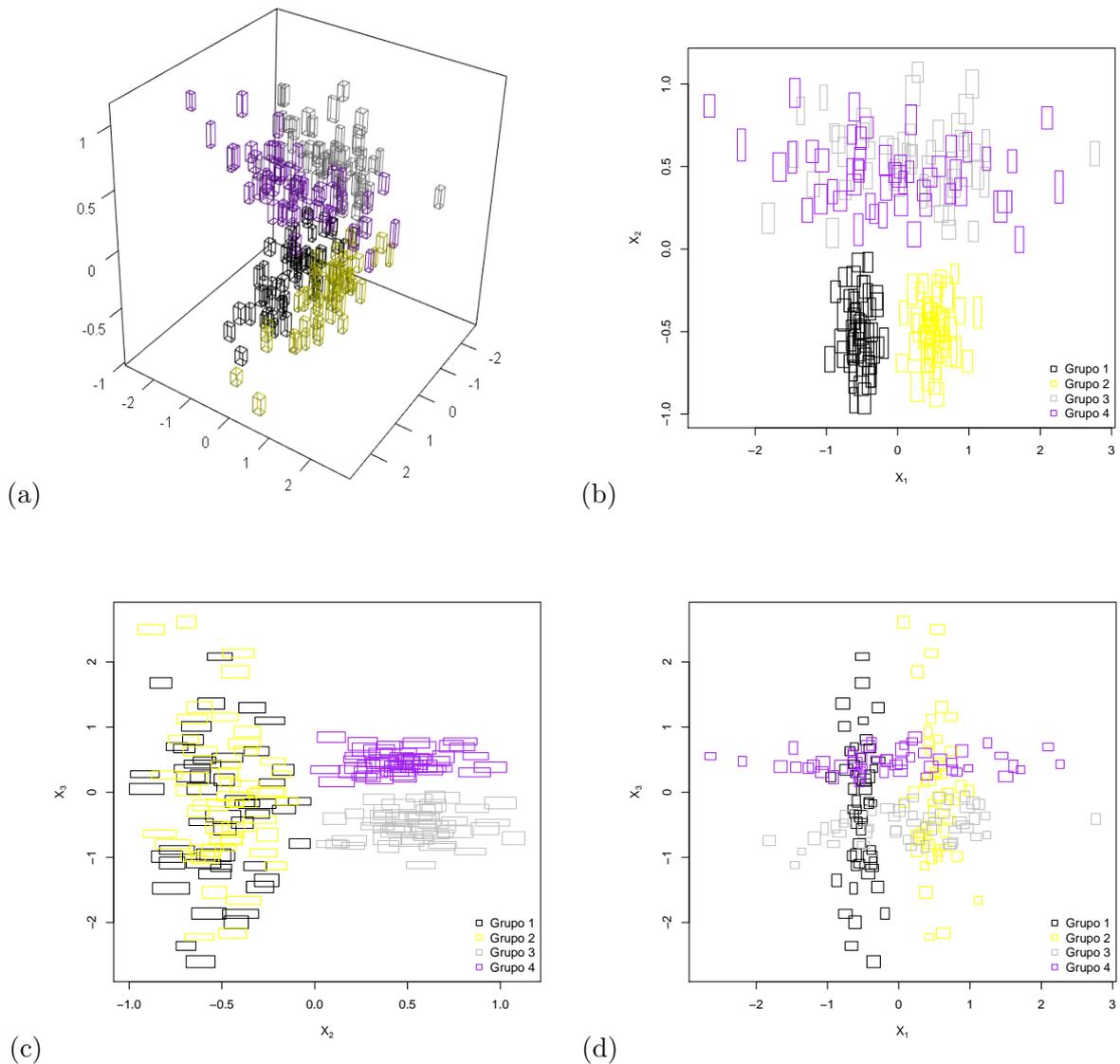


Figura 5.1: Conjunto de dados simulados 1 para o intervalo $[0.1, 0.2]$

Conjunto de dados simulados 2

O conjunto de dados simulados 2 é formado por 90 observações que estão divididos em três grupos de 30 . As variáveis x_1 e x_2 foram geradas a partir de uma distribuição normal bivariada com um vetor de médias específico para cada grupo e a mesma matriz de covariância para todos os grupos. Foi considerado que essas variáveis eram independentes. A variável x_3 foi gerada como uma combinação linear das duas primeiras mais um ruído aleatório gerado a partir de uma distribuição normal padrão, mais especificamente $x_{3i} = 2x_{1i} - 1.5x_{2i} + \varepsilon_i$, onde $\varepsilon_i \sim N(0, 1)$, $i = 1, \dots, n$. Devido a essa combinação linear, as variáveis x_1 e x_3 apresentam correlação positiva, enquanto as variáveis x_2 e x_3 são negativamente correlacionadas. A tabela 5.2 trás as informações das médias e variâncias usadas nessa configuração.

Tabela 5.2: Informações das estruturas de média e variância do conjunto de dados simulados 2.

μ	Grupo 1	Grupo 2	Grupo 3
μ_1	0,0	0,1	0,9
μ_2	0,0	0,7	0,1
Σ	Grupo 1	Grupo 2	Grupo 3
σ_1^2	0,04	0,04	0,04
σ_2^2	0,04	0,04	0,04

Observe pela figura 5.2 que todos os grupos apresentam o mesmo conjunto de variáveis importantes. Olhando para a figura 5.2 (b) percebe-se que as variáveis x_1 e x_2 são relevantes para a construção de todos os grupos, enquanto que a variável x_3 causa sobreposição para todos eles (figura 5.2 (c) e figura 5.2 (d)). Por fim, a mesma ressalva a respeito do intervalo para geração de γ feita no conjunto de dados simulados 1 se estende para esse caso.

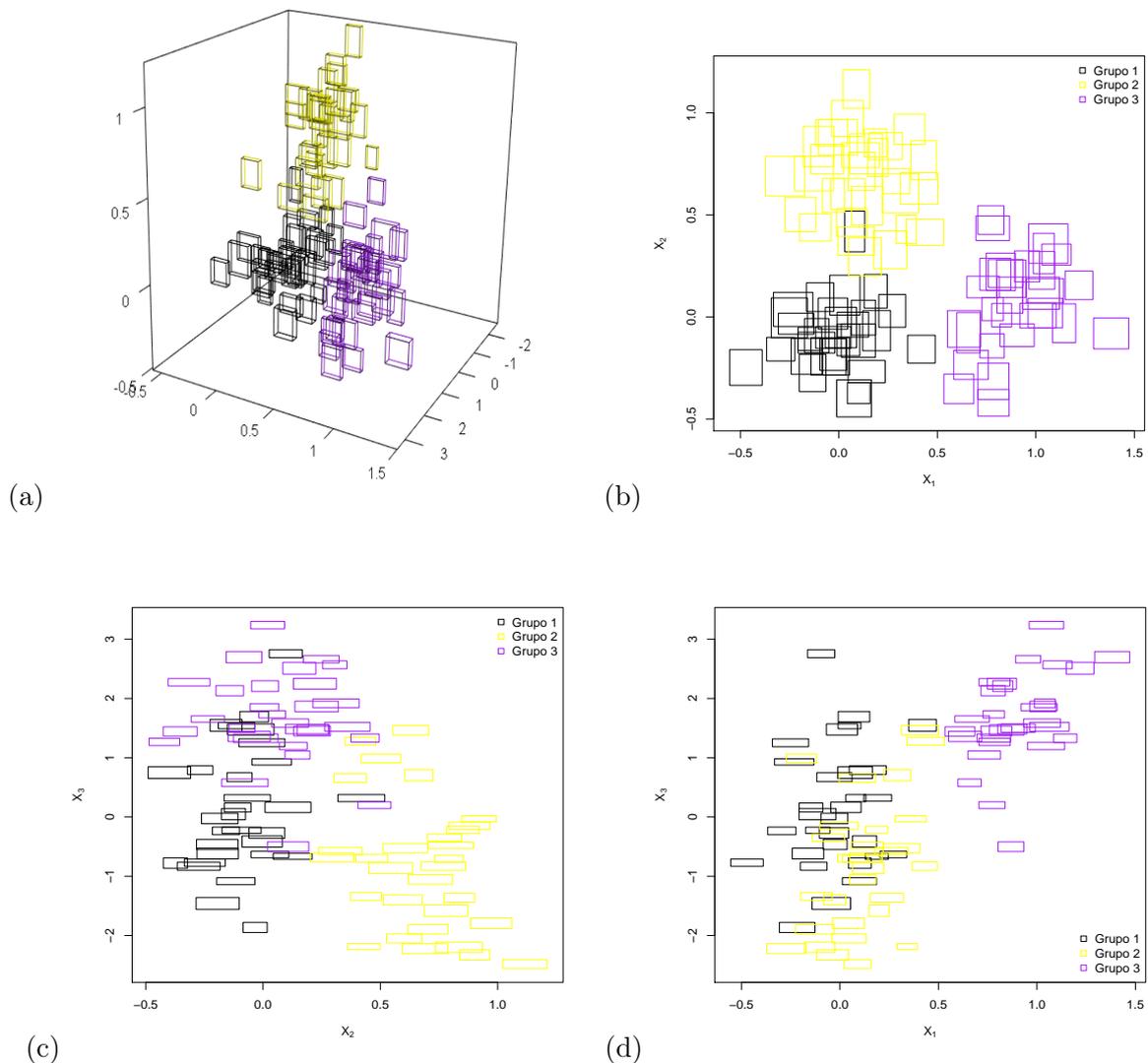


Figura 5.2: Conjunto de dados simulados 2 para o intervalo $[0.1, 0.2]$

5.1.2 Conjuntos de dados reais do tipo intervalo

Foram também realizados experimentos com doze conjuntos de dados reais do tipo intervalo, denominados aqui como *Autores*, *Fungos*, *Carros*, *Cavalos*, *Fórmula 1*, *Peixes*, *Tênis*, *Países*, *Amazon*, *NBA*, *Água* e *Temperatura*. Cada conjunto de dados é descrito a seguir.

Tabela 5.3: Dimensões dos conjuntos de dados reais.

Conjunto	n	p	K	Conjunto	n	p	K
<i>Autores</i>	21	5	2	<i>Tênis</i>	24	14	5
<i>Fungos</i>	55	5	3	<i>Países</i>	10	4	2
<i>Carros</i>	33	8	4	<i>Amazon</i>	25	4	3
<i>Cavalos</i>	15	4	4	<i>NBA</i>	29	5	3
<i>Fórmula 1</i>	10	9	2	<i>Água</i>	316	47	2
<i>Peixes</i>	12	13	4	<i>Temperatura</i>	37	12	4

Conjunto de dados: Autores

O conjunto de dados *Autores*¹ é resultado de uma pesquisa realizada pela *Université Paris Dauphine*. No contexto dessa pesquisa, os autores estavam interessados em analisar os livros que receberam algum prêmio, mais especificamente, eles realizaram a pesquisa se colocando no lugar de editoras com o intuito de analisar esses livros para obter informações relevantes que as norteariam em relação aos novos livros que estavam surgindo, principalmente seu potencial para ganhar prêmios e sua capacidade lucrativa.

Esse conjunto de dados trás informações a respeito de 21 premiações concedidas a livros, e para essa aplicação selecionamos as 5 variáveis intervalares disponíveis: nota do livro, data de nascimento do autor, ano da premiação, idade do autor quando ocorreu a premiação e número de livros de romances escritos pelo autor. Para a classe *a priori* utilizamos uma variável categórica disponível nesse conjunto de dados, que trás informações se o autor é ou não conhecidos pelos leitores.

Conjunto de dados: Fungos

O conjunto de dados simbólicos de natureza intervalar *Fungos*² trás informações a respeito de fungos encontrados na Califórnia, Estados Unidos. Os dados foram extraídos a partir de três gêneros de espécies de fungos: : Agaricus, Amanita e Boletus. De forma

¹www.ceremade.dauphine.fr/SODAS/EXEMPLES/choix_auteur_edition.htm

²<http://www.mykoweb.com/CAF/genera/>

sucinta, *Agaricus* é um gênero de cogumelos espinhosos, de médio a grande porte. Fazem parte desse gênero o *A. bisporus*, conhecido popularmente como o “cogumelo botão”, presente nos supermercados, e o *A. campestris*, o “cogumelo do prado”, comum em pastagens e prados e diversas regiões temperadas do mundo. *Amanita* também é um gênero de cogumelos espinhosos, de médio a grande porte. O membro mais famoso do gênero *Amanita* é o *A. muscaria*, o conhecido “fly agaric”, e famoso por causa do seu gorro vermelho adornado por verrugas brancas na parte superior. Já a principal característica do gênero *Boletus* são os esporos tipicamente castanho a castanho-oliva.

Esse conjunto de dados intervalares é composto por 55 observações de espécies, onde 16 são do gênero *Amanita*, 24 do gênero *Agaricus* e 15 do gênero *Boletus*. As espécies são descritas por cinco variáveis do tipo intervalo: largura do píleo, largura do estipe, espessura do estipe, altura dos poros e largura dos poros.

Conjunto de dados: Carros

O conjunto de dados simbólicos *Carros*³ é composto de 33 observações relacionadas a modelos de carros, e é descrito por oito variáveis simbólicas do tipo intervalo e uma variável nominal. As variáveis simbólicas desse conjunto são: preço, cilindrada do motor, velocidade máxima, aceleração, passo, comprimento, largura e altura. A variável nominal, denominada categoria de carro, foi considerada como sendo a classe *a priori*. Essa variável é formada por quatro grupos, sendo eles: Ammiraglia, Berlina, Sportiva e Utilitaria, compostos por 8,8,7 e 10 elementos, respectivamente.

Conjunto de dados: Cavalos

O conjunto de dados *Cavalos*⁴ é resultado de uma pesquisa realizada pela *Université Paris Dauphine*. Nessa ocasião os pesquisadores estavam interessados em descobrir se determinadas características dos cavalos (vestuário, categoria, aptidão) são agrupadas principalmente em um grupo de países e se esses países fazem parte de uma área bem demarcada. Esse banco é composto de 15 países e oito variáveis, das quais quatro delas são variáveis do tipo intervalo: nascimento, número de normas, peso máximo e tamanho máximo do torniquete. Para a classe *a priori* utilizamos os tipos de cavalo mais presente nos países. Essa categoria é formada por quatro grupos: *raça*, *lazer*, *ordenha* e *ponêi*.

³Oliveira et al. (2018)

⁴<https://www.ceremade.dauphine.fr/SODAS/EXEMPLES/chevaux2.htm>

Conjunto de dados: Fórmula 1

O conjunto de dados intervalares *Fórmula 1*⁵ é composto por 10 observações referentes a equipes de corrida da Fórmula 1. Esse conjunto de dados é descrito por 10 variáveis, sendo 9 variáveis simbólicas do tipo intervalo e uma variável nominal. As variáveis simbólicas são : Posição, Número de pontos, Ano de nascimento, Altura, Peso, Ano de início na F1, Número de grandes prêmios (GP), Poles, Número de vitórias em 2005. A variável nominal refere-se a marca de fabricante de pneus dos carros das equipes, que nesse caso po ser: michelin e bridgestone, com 3 e 7 observações cada marca, respectivamente.

Conjunto de dados: Peixes

O conjunto de dados simbólicos de natureza intervalar *Peixes*⁶ é composto por 12 observações relacionadas a espécies de peixes de água doce. Essas espécies são descritas por 13 variáveis intervalares, sendo elas: Comprimento, Peso, Músculo, Intestino, Estômago, Brânquias, Fígado, Rins, Fígado/músculo, Rins/músculo, Brânquias/músculo, Intestino/músculo e Estômago/músculo. As espécies estão agrupadas em quatro classes *a priori*: Carnivorous, Detritivorous, Omnivorous e Herbivorous, de modo que as duas primeiras classes tem tamanho 4 e as duas últimas tem tamanho 2.

Esse conjunto de dados é fruto de uma pesquisa realizada pelo Laboratoire d'Ecophysiologie et d'Ecotoxicologie des Systemes Aquatiques (LEESA), com intuito de buscar informações a respeito de uma contaminação atípica de mercúrio em algumas regiões da Guyana Francesa (COSTA, 2011).

Conjunto de dados: Tênis

O conjunto de dados *Tênis*⁷ é relacionado com os 50 melhores jogadores de tênis do mundo, segundo a classificação de dezembro de 2002. Esse conjunto de dados é composto por 24 observações referentes as nacionalidades dos jogadores. As variáveis simbólicas usadas nessa aplicação foram as seguintes: classificação no ATP de 2002, pontos no ATP, números de torneios jogados em 2002, idade, altura (cm), peso (Kg), ano de profissionalização, número de vitórias em 2002, número de derrotas em 2002, número de títulos em 2002, ganhos, número de vitórias difíceis, número de vitórias em terra, número de vitórias em grama, e número de vitórias em tapete. Para classe a priori consideramos os

⁵https://www.ceremade.dauphine.fr/SODAS/EXEMPLES/formule_1.htm

⁶<https://lhedjazi.jimdo.com/useful-links/>

⁷https://www.ceremade.dauphine.fr/SODAS/EXEMPLES/50_meilleurs_tennis.htm

continentes de nacionalização dos jogadores, que nessa ocasião foram: África, América, Ásia, Europa e Oceania.

Conjunto de dados: Países

O conjunto de dados intervalares *Países*⁸ é formado é composto por informações de 10 países que são descritos por 4 variáveis simbólica intervalar e uma variável nominal. As variáveis simbólicas do tipo intervalo são: população, expectativa de vida, Produto Interno Bruto (PIB) e Índice de Desenvolvimento Humano (IDH). A variável nominal refere-se a o desenvolvimento do país, onde 5 países são desenvolvidos e 5 países estão em desenvolvimento. Esta ultima variável foi usada para a classificação *a priori*.

Conjunto de dados: Amazon

O conjunto de dados *Amazon*⁹ é fruto de uma pesquisa realizada pela *Université Paris Dauphine*. Nessa pesquisa buscaram obter informações a respeito de quais categorias de livros vendidos pela Amazon recebem os melhores ranking de vendas, como também quais são suas características. Associaram a cada categoria um tipo de livro mais geral. Por exemplo, livros das categorias arte, música e cinema pertenciam ao tipo cultura. Já os livros da categorias quadrinhos e humor são parte do tipo lazer.

Esse conjunto é formado por 25 categorias de livros e é descrito por oito variáveis, onde 4 são variáveis simbólicas do tipo intervalo. As variáveis intervalares são : data de criação do livro, preço, ranking de venda e número de páginas. Para classe *a priori* consideramos a variável tipo, que é formada por três grupos: romance, cultura e lazer.

Conjunto de dados: NBA

O conjunto de dados *NBA*¹⁰ é fruto de uma pesquisa da *Université Paris Dauphine* que trás informações a respeito de 29 equipes da liga de basquete americana. A principal fonte de informação usada pela universidade foram os sites <www.sports.fr> e <www.nba.com>, recorrendo a outras fontes, como Yahoo e Google, quando não encontrava as informações necessárias nos dois primeiros. Esse conjunto de dados é formado por diversas variáveis e para o uso nesse trabalho consideramos as cinco variáveis intervalares

⁸Oliveira et al. (2018)

⁹https://www.ceremade.dauphine.fr/SODAS/EXEMPLES/ventes_livres_AMAZON.htm

¹⁰https://www.ceremade.dauphine.fr/SODAS/EXEMPLES/basket_NBA.htm

presentes: idade, altura (cm), peso (Kg), salário (milhões) e pontos feitos pelos jogadores. Além dessas, utilizamos um variável categórica, que trás a classificação do time ao final da liga (Boa, razoável e ruim) como classe *a priori*.

Conjunto de dados: Água

O conjunto de dados intervalares *Água*¹¹ trás informações a respeito de fluxos de água durante 30 minutos coletados em um período de 1 ano na rede de distribuição de água da cidade de Barcelona na Espanha. Apesar das observações terem sido durante um ano, o conjunto é composto por apenas 316 dias, pois em alguns dias os valores coletados foram descartados devido a problemas técnicos nos sensores de coleta. Essas informações são descritas por 48 variáveis intervalares que descrevem os valores mínimos e máximos de três medições consecutivas de 10 minutos do fluxo de água. Existem ainda uma outra variável categórica que separa os dias em dia útil e não útil (sábados, domingos e feriados). Para essa aplicação retiramos a variável 17 por se tratar de uma combinação das variáveis 15 e 16.

Conjunto de dados: Temperatura

Este conjunto de dados foi inicialmente apresentado em Guru, Kiranagi e Nagabhushan (2004), sendo composto por informações de temperaturas de 37 cidades. Essas cidades são descritas por 12 variáveis simbólicas do tipo intervalo sendo elas referentes as temperaturas mínima e máxima de cada mês do ano, medidas em graus centígrados. As cidade estão agrupadas em 4 classes *a priori* de tamanhos diferentes, onde a classe 1 é composta por 15 cidades, a classe 2 por 20 cidades e as classes 3 e 4 são composta por uma cidade.

5.2 Índices de avaliação

Para fazer a comparação dos métodos abordados nesse trabalho utilizamos o Índice Corrigido de Rand (CR) (HUBERT; ARABIE, 1985) e a Taxa Total de Erro de Classificação (OERC) (BREIMAN et al., 1984).

Considere $\mathcal{P} = \{\mathcal{P}_1, \dots, \mathcal{P}_i, \dots, \mathcal{P}_c\}$ a partição *a priori* de $\Omega = \{1, 2, \dots, n\}$ em c classes e seja $P = \{P_1, \dots, P_k, \dots, P_K\}$ uma partição de $\Omega = \{1, 2, \dots, n\}$ em K grupos

¹¹<https://lhedjazi.jimdo.com/useful-links/>

fornecida por um método de agrupamento. Considerando que as quantidades $n_{i,k}$, $i = 1 \dots, c$, $k = 1, \dots, K$ representam o número de observações que pertence a classe \mathcal{P}_i e ao grupo P_k , é possível representar - las através de uma matriz de confusão (Tabela 5.4).

Tabela 5.4: Matriz de confusão

Classes	Grupos					Σ
	P_1	\dots	P_k	\dots	P_K	
\mathcal{P}_1	n_{11}	\dots	n_{1k}	\dots	n_{1K}	$n_{1\bullet} = \sum_{k=1}^K n_{1k}$
\vdots	\vdots	\dots	\vdots	\dots	\vdots	\vdots
\mathcal{P}_i	n_{i1}	\dots	n_{ik}	\dots	n_{iK}	$n_{i\bullet} = \sum_{k=1}^K n_{ik}$
\vdots	\vdots	\dots	\vdots	\dots	\vdots	\vdots
\mathcal{P}_c	n_{c1}	\dots	n_{ck}	\dots	n_{cK}	$n_{c\bullet} = \sum_{k=1}^K n_{ck}$
Σ	$n_{\bullet 1} = \sum_{i=1}^c n_{i1}$	\dots	$n_{\bullet k} = \sum_{i=1}^c n_{ik}$	\dots	$n_{\bullet K} = \sum_{i=1}^c n_{iK}$	$n = \sum_{i=1}^c \sum_{k=1}^K n_{ik}$

O Índice Corrigido de Rand é obtido como

$$CR = \frac{\sum_{i=1}^c \sum_{k=1}^K \binom{n_{ik}}{2} - \binom{n}{2}^{-1} \sum_{i=1}^c \binom{n_{i\bullet}}{2} \sum_{k=1}^K \binom{n_{\bullet k}}{2}}{\frac{1}{2} \left[\sum_{i=1}^c \binom{n_{i\bullet}}{2} + \sum_{k=1}^K \binom{n_{\bullet k}}{2} \right] - \binom{n}{2}^{-1} \sum_{i=1}^c \binom{n_{i\bullet}}{2} \sum_{k=1}^K \binom{n_{\bullet k}}{2}}, \quad (5.1)$$

onde $n_{i\bullet}$ representa o número de padrões na classe \mathcal{P}_i , $n_{\bullet k}$ representa o número de observações no grupo P_k e n o total de observações na base de dados.

O Índice Corrigido de Rand avalia o grau de similaridade entre uma partição *a priori* e uma partição obtida por meio de um método de agrupamento. O índice CR assume valores no intervalo $[-1,1]$, de modo que o 1 indica similaridade perfeita entre as partições, enquanto que valores negativos ou próximos de zero indicam similaridade ao acaso (MILLIGAN; COOPER, 1986).

Em problemas de classificação, cada grupo P_k é associado a uma classe *a priori* \mathcal{P}_i como se esta fosse a verdadeira classe *a priori*, ou seja, se um padrão pertence a P_k a decisão está correta se sua *a priori* é \mathcal{P}_i . Dessa forma, para que a taxa de erro de classificação seja mínima, é necessário encontrar uma regra de decisão tal que a probabilidade de erro seja minimizada.

Seja $\ell(\mathcal{P}_i, P_k)$ a probabilidade *a posteriori* de que um padrão pertença à classe \mathcal{P}_i quando associado com o grupo P_k . Seja $\ell(P_k)$ a probabilidade de que um padrão pertença ao grupo P_k , onde ℓ é chamada de função de verossimilhança.

A estimativa da máxima probabilidade *a posteriori* é a moda da probabilidade *a*

posteriori $\ell(\mathcal{P}_i, P_k)$ e o índice da classe *a priori* associado a esta moda é :

$$MAP(P_k) = \arg \max_{1 \leq i \leq c} \ell(\mathcal{P}_i, P_k).$$

A regra de decisão de Bayes que minimiza a probabilidade média de erro é selecionar a classe *a priori* que maximiza a probabilidade *a posteriori*. A taxa de erro de classificação $ERC(P)$ do grupo P_k é igual a $1 - \ell(\mathcal{P}_{MAP(P_k)}/P_k)$ e a Taxa Total de Erro de Classificação (OERC) é dada por:

$$OERC = \sum_{k=1}^K \ell(P_k)(1 - \ell(\mathcal{P}_{MAP(P_k)}/P_k)).$$

Para uma amostra,

$$\ell(\mathcal{P}_{MAP(P_k)}/P_k) = \max_{1 \leq i \leq c} n_{ik}/n_{\bullet k}$$

A Taxa Total de Erro de Classificação foi idealizada para medir a habilidade que um método de agrupamento tem de encontrar classes *a priori* presente em um conjunto de dados. O índice OERC é dado por:

$$OERC = \sum_{k=1}^K \frac{n_{\bullet k}}{n} \left(1 - \max_{1 \leq i \leq c} n_{ik}/n_{\bullet k} \right) = 1 - \frac{\sum_{k=1}^K \max_{1 \leq i \leq c} n_{ik}}{n}. \quad (5.2)$$

Nesse caso, o índice OERC assume valores no intervalo $[0,1]$, no qual valores próximos de zero indicam maior capacidade do método em identificar classes *a priori*.

5.3 Resultados

Os algoritmos de agrupamentos discutidos nesse trabalho foram aplicados aos conjuntos de dados sintéticos e reais descritos anteriormente. Em cada aplicação foram calculados os índices CR e OERC e através destes ocorreu a comparação entre os métodos.

5.3.1 Conjunto de dados simulados do tipo intervalos

Para os conjuntos de dados sintéticos foi realizado um experimento de Monte Carlo com 100 réplicas. Em cada réplica os algoritmos de agrupamento foram executados 100 vezes (até a convergência, que ocorre quando a partição fornecida não se altera) e foi selecionado o melhor valor segundo os índices CR e OERC. O número de grupos foi definido como sendo o número de grupos *a priori* em cada conjunto de dados simulados. Para as 100 réplicas de Monte Carlo foi calculados a média e o desvio padrão dos índices CR e OERC. Para os algoritmos de agrupamento com a restrição que a soma dos pesos das variáveis é igual a um (iKKMEE-GS1, iKKMEE-LS1, iKKMEC-GS1, iKKMEC-LS1, iKKMEE-GS2, iKKMEE-LS2, iKKMEC-GS2 e iKKMEC-LS2) foi testado para o parâmetro θ os valores 2, 4 e 6. Em relação ao conjunto de dados simulados 1, $\theta = 6$ foi melhor no caso em que o intervalo de geração de γ foi $[0.1, 0.2]$ e $[0.1, 1.5]$, enquanto que para o intervalo $[1, 5]$ $\theta = 4$ apresentou melhor performance. Já para o conjunto de dados simulados 2, $\theta = 4$ foi melhor no caso em que o intervalo de geração de γ foi $[0.1, 0.2]$ e $[1, 5]$, enquanto que para o intervalo $[0.1, 1.5]$ $\theta = 2$ se sobressaiu. O termo σ^2 das funções *kernel* foi calculado como a médias entre os quantis 0,1 e 0,9 de $\|\mathbf{x}_i - \mathbf{x}_h\|^2$, $i \neq h$ segundo Caputo et al. (2001).

A tabela 5.5 apresenta o resultado dos índices CR e OERC para os algoritmos de agrupamento convencionais (iKM, iKKmeansEE e iKKmeansEC) e para os métodos de agrupamento *hard* baseados em *kernel* com ponderação automática das variáveis para o conjunto de dados simulados 1. A performance dos algoritmos baseado em distâncias adaptativas locais (iKKMEE-LP1, iKKMEE-LP2, iKKMEE-LS1, iKKMEE-LS2, iKKMEC-LP1, iKKMEC-LP2, iKKMEC-LS1 e iKKMEC-LS2) foi superior aos demais algoritmos em todas as situações, o que já era esperado, haja vista que nesse cenário o conjunto de variáveis importantes era diferente para cada grupo. Entre os métodos propostos, os que são baseados em distâncias não-adaptativas, mas com funções *kernel* separadas (iKKMEE1, iKKMEE2, iKKMEC1 e iKKMEC2), apresentaram piores resultados. Todos os métodos baseados em distâncias adaptativas apresentaram desempenho superior aos métodos convencionais, demonstrando a eficiência desses algoritmos. Cabe observar também que quando o intervalo de geração de γ é muito grande ($[1, 5]$) todos os algoritmos apresentam resultados piores, devido a sobreposição dos dados e conseqüentemente a mal definição dos grupos, mas ainda assim os algoritmos com ponderação automática das variáveis apresentaram desempenhos superiores.

Tabela 5.5: Resultado dos algoritmos de agrupamento segundo os índices CR e OERC para o conjunto de dados simulados 1 : média e desvio padrão (entre parênteses).

	[0.1, 0.2]		[0.1, 1.5]		[1, 5]	
	CR	OERC	CR	OERC	CR	OERC
iKM	0,3734 (0,0407)	0,3780 (0,0317)	0,3771 (0,0471)	0,3748 (0,0339)	0,2869 (0,0417)	0,4330 (0,0318)
iKMeansEE	0,3969 (0,0726)	0,3621 (0,0491)	0,4012 (0,0788)	0,3602 (0,0522)	0,2857 (0,0411)	0,4325 (0,0342)
iKMeansEC	0,4198 (0,0859)	0,3458 (0,0545)	0,4198 (0,0799)	0,3455 (0,0509)	0,2915 (0,0427)	0,4282 (0,0358)
iKMEE1	0,4555 (0,1078)	0,3313 (0,0663)	0,4441 (0,1059)	0,3391 (0,0643)	0,2870 (0,0433)	0,4302 (0,034)
iKMEE2	0,4115 (0,0785)	0,3551 (0,0511)	0,4034 (0,0679)	0,3606 (0,0464)	0,2856 (0,0426)	0,4324 (0,0364)
iKMEE-GP1	0,6581 (0,0265)	0,2208 (0,0255)	0,6586 (0,0299)	0,2197 (0,0269)	0,3558 (0,0473)	0,4025 (0,0418)
iKMEE-GP2	0,6535 (0,0144)	0,2246 (0,0188)	0,6531 (0,0154)	0,2233 (0,019)	0,3484 (0,0389)	0,4110 (0,037)
iKMEE-GS1	0,6537 (0,0138)	0,2239 (0,0179)	0,6552 (0,0136)	0,2220 (0,019)	0,3637 (0,0643)	0,4038 (0,0506)
iKMEE-GS2	0,6527 (0,0151)	0,2248 (0,0187)	0,6540 (0,0139)	0,2241 (0,0186)	0,3520 (0,0457)	0,4179 (0,0417)
iKMEE-LP1	0,9770 (0,0163)	0,0087 (0,0061)	0,974 (0,0176)	0,0098 (0,0066)	0,4831 (0,0957)	0,3070 (0,0569)
iKMEE-LP2	0,9771 (0,0161)	0,0086 (0,0061)	0,9729 (0,0183)	0,0102 (0,0069)	0,4694 (0,0949)	0,3176 (0,0627)
iKMEE-LS1	0,9768 (0,0162)	0,0087 (0,0061)	0,9746 (0,0176)	0,0096 (0,0067)	0,5557 (0,0878)	0,2683 (0,0487)
iKMEE-LS2	0,9759 (0,0169)	0,0091 (0,0064)	0,9746 (0,0176)	0,0096 (0,0066)	0,5255 (0,0843)	0,2863 (0,0498)
iKMEC1	0,5694 (0,1072)	0,2668 (0,0683)	0,5489 (0,1178)	0,2787 (0,0737)	0,3150 (0,047)	0,4135 (0,0343)
iKMEC2	0,4642 (0,1037)	0,3261 (0,0595)	0,4548 (0,1026)	0,3336 (0,0619)	0,2914 (0,0407)	0,4280 (0,0341)
iKMEC-GP1	0,6714 (0,0641)	0,2104 (0,0475)	0,6922 (0,0934)	0,1937 (0,0673)	0,3741 (0,0707)	0,3880 (0,0525)
iKMEC-GP2	0,6626 (0,0494)	0,2169 (0,0387)	0,6636 (0,0499)	0,2161 (0,0386)	0,3548 (0,0506)	0,4011 (0,0412)
iKMEC-GS1	0,6601 (0,0337)	0,2200 (0,0277)	0,6712 (0,0601)	0,2099 (0,0457)	0,3768 (0,0691)	0,3892 (0,0494)
iKMEC-GS2	0,6567 (0,0326)	0,2219 (0,0277)	0,6614 (0,0422)	0,2178 (0,0334)	0,3564 (0,0581)	0,4060 (0,0437)
iKMEC-LP1	0,9774 (0,0164)	0,0085 (0,0062)	0,9746 (0,0177)	0,0096 (0,0067)	0,5591 (0,1052)	0,2620 (0,0677)
iKMEC-LP2	0,9778 (0,0158)	0,0084 (0,0059)	0,9739 (0,0174)	0,0098 (0,0065)	0,4948 (0,0964)	0,3014 (0,0576)
iKMEC-LS1	0,9764 (0,0164)	0,0089 (0,0062)	0,9751 (0,0175)	0,0094 (0,0066)	0,5775 (0,0873)	0,2480 (0,0549)
iKMEC-LS2	0,9763 (0,0168)	0,0089 (0,0063)	0,9751 (0,0172)	0,0094 (0,0065)	0,5407 (0,0943)	0,2719 (0,0569)

Tabela 5.6: Resultado dos algoritmos de agrupamento segundo os índices CR e OERC para o conjunto de dados simulados 2 : média e desvio padrão (entre parênteses).

	[0.1, 0.2]		[0.1, 1.5]		[1, 5]	
	CR	OERC	CR	OERC	CR	OERC
iKM	0,2679 (0,0605)	0,3509 (0,0487)	0,2702 (0,0631)	0,3497 (0,0481)	0,2909 (0,0589)	0,3373 (0,0436)
iKMeansEE	0,2827 (0,0665)	0,3404 (0,053)	0,2831 (0,0686)	0,3401 (0,0521)	0,2946 (0,0605)	0,3331 (0,0457)
iKMeansEC	0,2874 (0,0648)	0,3399 (0,049)	0,2874 (0,0637)	0,3407 (0,0492)	0,3010 (0,0580)	0,3294 (0,0436)
iKMEE1	0,2842 (0,0626)	0,3423 (0,0486)	0,2915 (0,0776)	0,3362 (0,0541)	0,2966 (0,0574)	0,3323 (0,0436)
iKMEE2	0,2831 (0,0754)	0,3420 (0,0538)	0,2796 (0,0647)	0,3429 (0,0510)	0,2937 (0,0593)	0,3339 (0,0448)
iKMEE-GP1	0,8908 (0,0631)	0,0377 (0,0227)	0,8472 (0,0865)	0,0538 (0,0325)	0,4024 (0,0717)	0,2704 (0,047)
iKMEE-GP2	0,8863 (0,0688)	0,0393 (0,0251)	0,8391 (0,0951)	0,0571 (0,0373)	0,4017 (0,0763)	0,2679 (0,0477)
iKMEE-GS1	0,9037 (0,0583)	0,0331 (0,0207)	0,9064 (0,0518)	0,0321 (0,0184)	0,4230 (0,0804)	0,2654 (0,0533)
iKMEE-GS2	0,9037 (0,0585)	0,0331 (0,0208)	0,9048 (0,0515)	0,0327 (0,0182)	0,4278 (0,0818)	0,2621 (0,0544)
iKMEE-LP1	0,8904 (0,0658)	0,0379 (0,0237)	0,8343 (0,0948)	0,0591 (0,0369)	0,4088 (0,0708)	0,2720 (0,0471)
iKMEE-LP2	0,8864 (0,0730)	0,0394 (0,0266)	0,8185 (0,1007)	0,0651 (0,0395)	0,4008 (0,0660)	0,2758 (0,0431)
iKMEE-LS1	0,9060 (0,0587)	0,0323 (0,0209)	0,9031 (0,0544)	0,0333 (0,0193)	0,4437 (0,0622)	0,2646 (0,0439)
iKMEE-LS2	0,9050 (0,0603)	0,0328 (0,0216)	0,9020 (0,0539)	0,0337 (0,0191)	0,4434 (0,0630)	0,2659 (0,0470)
iKMEC1	0,3115 (0,0561)	0,3371 (0,0445)	0,3107 (0,0602)	0,3360 (0,0465)	0,3051 (0,0603)	0,3268 (0,0446)
iKMEC2	0,3080 (0,0584)	0,3428 (0,0468)	0,3109 (0,0597)	0,3394 (0,047)	0,3016 (0,0608)	0,3312 (0,0458)
iKMEC-GP1	0,8921 (0,0641)	0,0372 (0,0233)	0,8494 (0,0885)	0,0531 (0,0333)	0,4176 (0,0762)	0,2600 (0,0493)
iKMEC-GP2	0,8883 (0,0661)	0,0385 (0,0239)	0,8396 (0,0878)	0,0567 (0,0332)	0,4074 (0,0762)	0,2650 (0,0473)
iKMEC-GS1	0,9034 (0,0580)	0,0332 (0,0206)	0,9067 (0,0521)	0,0320 (0,0184)	0,4397 (0,0863)	0,2541 (0,0549)
iKMEC-GS2	0,9024 (0,0583)	0,0335 (0,0207)	0,9064 (0,0526)	0,0321 (0,0186)	0,4357 (0,0844)	0,2558 (0,0550)
iKMEC-LP1	0,8900 (0,0675)	0,0381 (0,0245)	0,8346 (0,1021)	0,0596 (0,0419)	0,4155 (0,0698)	0,2674 (0,0463)
iKMEC-LP2	0,8871 (0,0692)	0,0391 (0,0252)	0,8212 (0,1034)	0,0642 (0,0406)	0,4093 (0,0711)	0,2680 (0,0452)
iKMEC-LS1	0,9064 (0,0577)	0,0322 (0,0206)	0,9023 (0,0558)	0,0337 (0,0200)	0,4525 (0,0608)	0,2610 (0,0457)
iKMEC-LS2	0,9060 (0,0588)	0,0323 (0,0210)	0,9016 (0,0559)	0,0339 (0,0199)	0,4419 (0,0601)	0,2649 (0,0450)

A tabela 5.6 apresenta a performance dos algoritmos de agrupamento convencionais (iKM, iKKmeansEE e iKKmeansEC) e dos métodos de agrupamento *hard* baseados em *kernel* com ponderação automática das variáveis para o conjunto de dados simulados 2, segundo os índices CR e OERC. O desempenho dos algoritmos baseados em distâncias adaptativas locais e globais com a restrição de que a soma dos pesos das variáveis deve ser igual a um (iKKMEE-GS1, iKKMEE-GS2, iKKMEE-LS1, iKKMEE-LS2, iKKMEC-GS1, iKKMEC-GS2, iKKMEC-LS1 e iKKMEC-LS2) foi superior aos demais algoritmos em todas as situações. Nesse cenário, os métodos baseados na distância adaptativa global apresentaram desempenho muito melhor quando comparado com o resultado do cenário 1, o que já era esperado, pois nesse cenário o conjunto de variáveis importantes era igual para todos os grupos. Entre os métodos propostos, os que são baseados em distâncias não-adaptativas, mas com funções *kernel* separadas (iKKMEE1, iKKMEE2, iKKMEC1 e iKKMEC2) apresentaram piores resultados. Todos os métodos baseados em distâncias adaptativas apresentaram desempenho superior aos métodos convencionais, demonstrando a eficiência desses algoritmos. Mais uma vez, quando o intervalo de geração de γ é muito grande ($[1, 5]$) todos os algoritmos baseados em distâncias adaptativas apresentam resultados piores, devido a sobreposição dos dados e a mal definição dos grupos, mas ainda assim considerar pesos para as variáveis durante o processo de agrupamento é de grande importância, visto que os métodos baseado com ponderação automáticas apresentaram desempenho superior.

Inicialmente, para comparar o desempenho dos algoritmos de agrupamento baseado em distâncias adaptativas segundo as funções *kernel* de uma e duas componentes, foi realizado nas 100 amostras das Simulações de Monte Carlo um teste de normalidade de Shapiro-Wilk para verificar se nas amostras os índices CR e OERC seguiam normalidade, e assim realizar um teste t pareado. Considerando 5% de significância, rejeitamos a hipótese de normalidade nas amostras, e para fazer a comparação foi realizado testes não paramétrico de Wilcoxon pareado. As tabelas 5.7 e 5.8 apresentam esses resultados considerando 5% de significância. Nessas tabelas, o símbolo “=” significa que, estatisticamente, os algoritmos não apresentam diferenças segundo o parâmetro de locação em relação aquele determinado índice. O símbolo “+” significa que, estatisticamente, o algoritmo que usa o *kernel* de uma componente apresenta parâmetro de locação superior ao algoritmo com o *kernel* de duas componentes, em relação aquele determinado índice. Finalmente, o símbolo “-” significa que, estatisticamente, o algoritmo que usa o *kernel* de uma componente apresenta parâmetro de locação inferior ao algoritmo com o *kernel* de duas componentes, em relação aquele determinado índice.

Tabela 5.7: Resultado do teste de Wilcoxon pareado nas amostras do conjunto de dados simulados 1 para comparar os métodos segundo as funções *kernel*.

	CR			OERC		
	[0.1, 0.2]	[0.1, 1.5]	[1, 5]	[0.1, 0.2]	[0.1, 1.5]	[1, 5]
iKKMEE-GP	+	+	=	-	-	-
iKKMEE-LP	=	=	=	=	=	=
iKKMEC-GP	+	+	+	-	-	-
iKKMEC-LP	=	=	+	=	=	-
iKKMEE-GS	+	+	+	=	-	-
iKKMEE-LS	=	=	+	=	=	-
iKKMEC-GS	+	+	+	-	-	-
iKKMEC-LS	=	=	+	=	=	-

Tabela 5.8: Resultado do teste de Wilcoxon pareado nas amostras do conjunto de dados simulados 2 para comparar os métodos segundo as funções *kernel*.

	CR			OERC		
	[0.1, 0.2]	[0.1, 1.5]	[1, 5]	[0.1, 0.2]	[0.1, 1.5]	[1, 5]
iKKMEE-GP	+	+	=	-	-	=
iKKMEE-LP	=	+	+	=	-	-
iKKMEC-GP	+	+	+	-	-	-
iKKMEC-LP	=	+	+	=	-	=
iKKMEE-GS	=	=	=	=	=	=
iKKMEE-LS	=	=	=	=	=	=
iKKMEC-GS	=	=	=	=	=	=
iKKMEC-LS	=	=	+	=	=	=

Considerando que para o índice CR maiores valores indicam melhores resultados, enquanto que para o índice OERC menores valores indicam melhores resultados, ao analisar as tabelas 5.7 e 5.8 podemos concluir que, estatisticamente, os algoritmos de agrupamento que usam o *kernel* de uma componente apresentaram desempenhos iguais ou superiores quando comparado com os algoritmos de usam o *kernel* de duas componentes. Para o conjunto de dados simulados 1 os algoritmos baseado nas distâncias adaptativas globais com o *kernel* de uma componente (iKKMEE-GP1, iKKMEC-GP1, iKKMEE-GS1 e iKKMEC-GS1) foram, em quase todas as situações, superiores a suas versões com o *kernel* de duas componentes, enquanto que para o conjunto de dados simulados 2 destaca - se apenas o algoritmo iKKMEC-GP1.

5.3.2 Conjunto de dados reais do tipo intervalo

Os algoritmos de agrupamentos convencionais para dados intervalares: K -médias (iKM), *kernel* K -médias na kernelização da métrica (iKKmeansEE) e *kernel* K -médias no espaço de características (iKKmeansEC), e os algoritmos propostos nesse trabalho foram

aplicados aos 12 conjuntos de dados reais intervalares apresentados na seção 5.1.2. Para cada conjunto de dados os algoritmos foram executados 100 vezes (até a convergência, que ocorre quando a partição fornecida pelo algoritmo não se altera mais) e foi selecionado o melhor resultado segundo o Índice Corrigido de Rand (CR) e a Taxa Total de Erro de Classificação (OERC). O número de grupos considerados foi igual ao número de grupos em cada classe *a priori*, conforme apresentado na descrição dos bancos de dados na seção 5.1.2. O termo σ^2 das funções de *kernel* foi calculado da mesma forma que na avaliação para os dados simulados. Os algoritmos sob a restrição de que a soma dos pesos deve ser igual a um (iKKMEE-GS1, iKKMEE-LS1, iKKMEC-GS1, iKKMEC-LS1, iKKMEE-GS2, iKKMEE-LS2, iKKMEC-GS2 e iKKMEC-LS2) foram executados para $\theta = 2, 4, 6$. O conjunto de dados *Cavalos* apresentou melhor resultado para $\theta = 2$. Os conjuntos *Fórmula 1*, *Peixes*, *Tênis*, *Amazon*, *Água* e *Temperatura* apresentaram melhor resultado para $\theta = 4$, enquanto os conjuntos *Autores*, *Fungos*, *Carros*, *Países* e *NBA* apresentaram melhor resultado para $\theta = 6$.

As tabelas 5.9 e 5.10 apresentam os índices CR e OERC, respectivamente, calculados entre as partições fornecidas pelos algoritmos e as classes *a priori* em cada conjunto de dados. Adicionalmente, entre parênteses, é apresentado o desempenho de cada algoritmo, segundo os respectivos critérios.

A tabela 5.11 mostra a classificação média e mediana dos algoritmos segundos os índices CR e OERC, calculados a partir dos resultados apresentados nas tabelas 5.9 e 5.10. Adicionalmente, entre parenteses, é apresentada a classificação dos métodos segundo o desempenho médio e mediano.

A tabela 5.12 apresenta o desempenho dos algoritmos segundo as funções *kernel* de uma e duas componentes, calculado a partir da tabela 5.11. Adicionalmente, entre parênteses, é apresentado sua classificação. Para construção dessa tabela, foi calculado a média entre o desempenho médio segundos os índices CR e OERC e a média entre o desempenho mediano segundo esses mesmos índices.

Tabela 5.9: Resultado do Índice Corrigido de Rand (CR) para os dados simbólicos intervalares reais considerados.

	<i>Autores</i>	<i>Fungos</i>	<i>Carros</i>	<i>Cavalos</i>	<i>Fórmula 1</i>	<i>Peixes</i>
iKM	0,0730 (23)	0,4547 (20)	0,3884 (19)	0,4964 (19)	-0,0714 (23)	0,1842 (18)
iKKmeansEE	0,3526 (16)	0,5248 (18)	0,4466 (17)	0,4964 (19)	0,2857 (9)	0,1842 (18)
iKKmeansEC	0,3526 (16)	0,6287 (17)	0,3884 (19)	0,4964 (19)	0,0708 (16)	0,0947 (22)
iKKMEE1	0,3526 (16)	0,4547 (20)	0,3884 (19)	0,4964 (19)	0,2857 (9)	0,2757 (17)
iKKMEE2	0,3526 (16)	0,5248 (18)	0,3884 (19)	0,4964 (19)	0,0708 (16)	0,1842 (18)
iKKMEE-GP1	1,0000 (1)	0,8291 (9)	0,7246 (1)	0,6887 (3)	0,5921 (1)	0,3295 (15)
iKKMEE-GP2	0,8093 (5)	0,8291 (9)	0,7246 (1)	0,6887 (3)	0,5921 (1)	0,3295 (15)
iKKMEE-GS1	1,0000 (1)	0,8291 (9)	0,6142 (11)	0,7099 (1)	0,5921 (1)	0,7673 (1)
iKKMEE-GS2	1,0000 (1)	0,8291 (9)	0,6142 (11)	0,7099 (1)	0,5921 (1)	0,7284 (6)
iKKMEE-LP1	0,4853 (12)	0,8837 (1)	0,6802 (6)	0,6887 (3)	0,2593 (12)	0,4272 (10)
iKKMEE-LP2	0,4853 (12)	0,8837 (1)	0,6802 (6)	0,5767 (13)	0,2593 (12)	0,4272 (10)
iKKMEE-LS1	0,8093 (5)	0,8837 (1)	0,7015 (4)	0,6281 (9)	0,2593 (12)	0,7284 (6)
iKKMEE-LS2	0,8093 (5)	0,8837 (1)	0,7015 (4)	0,6281 (9)	0,2593 (12)	0,7673 (1)
iKKMEC1	0,3526 (16)	0,4547 (20)	0,4161 (18)	0,5286 (16)	0,0708 (16)	0,0947 (22)
iKKMEC2	0,3526 (16)	0,4265 (23)	0,3884 (19)	0,5286 (16)	0,0708 (16)	0,1417 (21)
iKKMEC-GP1	0,8093 (5)	0,8291 (9)	0,6641 (8)	0,5364 (15)	0,2941 (5)	0,3473 (12)
iKKMEC-GP2	0,3526 (16)	0,8291 (9)	0,7246 (1)	0,6887 (3)	0,0708 (16)	0,521 (9)
iKKMEC-GS1	1,0000 (1)	0,8291 (9)	0,6142 (11)	0,5977 (12)	0,2941 (5)	0,7673 (1)
iKKMEC-GS2	0,8093 (5)	0,8291 (9)	0,6142 (11)	0,6854 (7)	0,0708 (16)	0,7673 (1)
iKKMEC-LP1	0,4853 (12)	0,8837 (1)	0,6249 (9)	0,5565 (14)	0,0708 (16)	0,3473 (12)
iKKMEC-LP2	0,4853 (12)	0,8837 (1)	0,6249 (9)	0,5286 (16)	0,2941 (5)	0,3473 (12)
iKKMEC-LS1	0,8093 (5)	0,8837 (1)	0,6142 (11)	0,6854 (7)	0,2857 (9)	0,6567 (8)
iKKMEC-LS2	0,8093 (5)	0,8837 (1)	0,6142 (11)	0,6106 (11)	0,2941 (5)	0,7673 (1)
	<i>Tênis</i>	<i>Países</i>	<i>Amazon</i>	<i>NBA</i>	<i>Água</i>	<i>Temperatura</i>
iKM	0,0968 (20)	-0,0766 (23)	-0,0308 (18)	0,0174 (23)	0,0995 (6)	0,8800 (3)
iKKmeansEE	0,1149 (14)	0,0597 (20)	-0,0308 (18)	0,0641 (14)	0,0995 (6)	0,8800 (3)
iKKmeansEC	0,0783 (23)	0,0957 (18)	0,0000 (17)	0,0652 (12)	0,0995 (6)	0,6687 (23)
iKKMEE1	0,1442 (10)	0,2800 (1)	-0,0308 (18)	0,0448 (18)	0,0995 (6)	0,8800 (3)
iKKMEE2	0,1079 (15)	0,0957 (18)	-0,0308 (18)	0,0267 (22)	0,0995 (6)	0,8437 (11)
iKKMEE-GP1	0,1532 (9)	0,2800 (1)	0,1607 (11)	0,0611 (15)	0,0958 (15)	0,7119 (22)
iKKMEE-GP2	0,1384 (12)	0,2800 (1)	0,2034 (1)	0,0611 (15)	0,0958 (15)	0,7217 (18)
iKKMEE-GS1	0,1890 (8)	0,2800 (1)	0,1696 (7)	0,0652 (12)	0,0911 (20)	0,8800 (3)
iKKMEE-GS2	0,2165 (5)	0,2800 (1)	0,1696 (7)	0,162 (1)	0,0911 (20)	0,8800 (3)
iKKMEE-LP1	0,1057 (17)	0,2800 (1)	0,0273 (14)	0,0882 (4)	0,1005 (3)	0,8808 (2)
iKKMEE-LP2	0,1424 (11)	0,2800 (1)	0,1212 (12)	0,0699 (10)	0,1005 (3)	0,9110 (1)
iKKMEE-LS1	0,2501 (1)	0,2800 (1)	0,1722 (6)	0,0668 (11)	0,1113 (1)	0,8800 (3)
iKKMEE-LS2	0,2355 (2)	0,2800 (1)	0,1696 (7)	0,0407 (19)	0,109 (2)	0,8800 (3)
iKKMEC1	0,0875 (21)	0,0000 (22)	-0,0308 (18)	0,0351 (20)	0,0995 (6)	0,7215 (19)
iKKMEC2	0,1058 (16)	0,0597 (20)	-0,0308 (18)	0,0351 (20)	0,0995 (6)	0,8800 (3)
iKKMEC-GP1	0,0828 (22)	0,2800 (1)	0,0006 (16)	0,0717 (8)	0,0958 (15)	0,7215 (19)
iKKMEC-GP2	0,1012 (18)	0,2800 (1)	0,1681 (10)	0,0517 (17)	0,0958 (15)	0,7215 (19)
iKKMEC-GS1	0,2343 (3)	0,2800 (1)	0,1733 (3)	0,0848 (5)	0,0911 (20)	0,8197 (15)
iKKMEC-GS2	0,1987 (7)	0,2800 (1)	0,1733 (3)	0,0738 (7)	0,0911 (20)	0,7453 (16)
iKKMEC-LP1	0,0989 (19)	0,2800 (1)	0,0108 (15)	0,1029 (2)	0,1005 (3)	0,8209 (14)
iKKMEC-LP2	0,1282 (13)	0,2800 (1)	0,0758 (13)	0,1029 (2)	0,0995 (6)	0,8437 (11)
iKKMEC-LS1	0,2215 (4)	0,2800 (1)	0,1733 (3)	0,071 (9)	0,0958 (15)	0,8437 (11)
iKKMEC-LS2	0,2016 (6)	0,2800 (1)	0,1836 (2)	0,0749 (6)	0,0995 (6)	0,7299 (17)

Tabela 5.10: Resultado da Taxa Total de Erro de Classificação (OERC) para os dados simbólicos intervalares reais considerados.

	<i>Autores</i>	<i>Fungos</i>	<i>Carros</i>	<i>Cavalos</i>	<i>Fórmula 1</i>	<i>Peixes</i>
iKM	0,3333 (23)	0,2909 (21)	0,3333 (18)	0,2000 (11)	0,3000 (16)	0,3333 (17)
iKKmeansEE	0,1905 (16)	0,2545 (18)	0,3030 (17)	0,2000 (11)	0,2000 (5)	0,3333 (17)
iKKmeansEC	0,1905 (16)	0,1455 (17)	0,3333 (18)	0,2000 (11)	0,3000 (16)	0,4167 (22)
iKKMEE1	0,1905 (16)	0,2909 (21)	0,3333 (18)	0,2000 (11)	0,2000 (5)	0,2500 (10)
iKKMEE2	0,1905 (16)	0,2545 (18)	0,3333 (18)	0,2000 (11)	0,3000 (16)	0,3333 (17)
iKKMEE-GP1	0,0000 (1)	0,0545 (9)	0,1212 (1)	0,1333 (1)	0,1000 (1)	0,2500 (10)
iKKMEE-GP2	0,0476 (5)	0,0545 (9)	0,1212 (1)	0,1333 (1)	0,1000 (1)	0,2500 (10)
iKKMEE-GS1	0,0000 (1)	0,0545 (9)	0,1515 (8)	0,1333 (1)	0,1000 (1)	0,0833 (1)
iKKMEE-GS2	0,0000 (1)	0,0545 (9)	0,1515 (8)	0,1333 (1)	0,1000 (1)	0,0833 (1)
iKKMEE-LP1	0,1429 (12)	0,0364 (1)	0,1212 (1)	0,1333 (1)	0,2000 (5)	0,2500 (10)
iKKMEE-LP2	0,1429 (12)	0,0364 (1)	0,1212 (1)	0,2000 (11)	0,2000 (5)	0,2500 (10)
iKKMEE-LS1	0,0476 (5)	0,0364 (1)	0,1212 (1)	0,1333 (1)	0,2000 (5)	0,0833 (1)
iKKMEE-LS2	0,0476 (5)	0,0364 (1)	0,1212 (1)	0,1333 (1)	0,2000 (5)	0,0833 (1)
iKKMEC1	0,1905 (16)	0,2727 (20)	0,3333 (18)	0,2000 (11)	0,3000 (16)	0,4167 (22)
iKKMEC2	0,1905 (16)	0,2909 (21)	0,3333 (18)	0,2000 (11)	0,3000 (16)	0,3333 (17)
iKKMEC-GP1	0,0476 (5)	0,0545 (9)	0,1515 (8)	0,2000 (11)	0,2000 (5)	0,3333 (17)
iKKMEC-GP2	0,1905 (16)	0,0545 (9)	0,1212 (1)	0,1333 (1)	0,3000 (16)	0,1667 (8)
iKKMEC-GS1	0,0000 (1)	0,0545 (9)	0,1515 (8)	0,2000 (11)	0,2000 (5)	0,0833 (1)
iKKMEC-GS2	0,0476 (5)	0,0545 (9)	0,1515 (8)	0,1333 (1)	0,3000 (16)	0,0833 (1)
iKKMEC-LP1	0,1429 (12)	0,0364 (1)	0,1818 (15)	0,2000 (11)	0,3000 (16)	0,2500 (10)
iKKMEC-LP2	0,1429 (12)	0,0364 (1)	0,1818 (15)	0,2000 (11)	0,2000 (5)	0,2500 (10)
iKKMEC-LS1	0,0476 (5)	0,0364 (1)	0,1515 (8)	0,1333 (1)	0,2000 (5)	0,1667 (8)
iKKMEC-LS2	0,0476 (5)	0,0364 (1)	0,1515 (8)	0,2000 (11)	0,2000 (5)	0,0833 (1)
	<i>Tênis</i>	<i>Países</i>	<i>Amazon</i>	<i>NBA</i>	<i>Água</i>	<i>Temperatura</i>
iKM	0,4167 (14)	0,5000 (23)	0,5600 (17)	0,5172 (22)	0,2943 (1)	0,0541 (1)
iKKmeansEE	0,3750 (5)	0,3000 (18)	0,5600 (17)	0,4483 (5)	0,3323 (13)	0,0811 (8)
iKKmeansEC	0,3750 (5)	0,3000 (18)	0,5600 (17)	0,4828 (14)	0,3323 (13)	0,1892 (22)
iKKMEE1	0,3750 (5)	0,2000 (1)	0,5600 (17)	0,4828 (14)	0,3323 (13)	0,0811 (8)
iKKMEE2	0,3750 (5)	0,3000 (18)	0,5600 (17)	0,5172 (22)	0,2943 (1)	0,0541 (1)
iKKMEE-GP1	0,3333 (2)	0,2000 (1)	0,3600 (1)	0,4483 (5)	0,3354 (20)	0,1351 (17)
iKKMEE-GP2	0,3750 (5)	0,2000 (1)	0,3600 (1)	0,4828 (14)	0,3354 (20)	0,2162 (23)
iKKMEE-GS1	0,3750 (5)	0,2000 (1)	0,3600 (1)	0,4483 (5)	0,3133 (10)	0,0811 (8)
iKKMEE-GS2	0,3750 (5)	0,2000 (1)	0,3600 (1)	0,3793 (1)	0,2943 (1)	0,0811 (8)
iKKMEE-LP1	0,3750 (5)	0,2000 (1)	0,4400 (11)	0,4483 (5)	0,2943 (1)	0,0541 (1)
iKKMEE-LP2	0,3333 (2)	0,2000 (1)	0,4400 (11)	0,4483 (5)	0,2943 (1)	0,0541 (1)
iKKMEE-LS1	0,3333 (2)	0,2000 (1)	0,3600 (1)	0,4828 (14)	0,2943 (1)	0,0811 (8)
iKKMEE-LS2	0,2917 (1)	0,2000 (1)	0,3600 (1)	0,4828 (14)	0,2943 (1)	0,0811 (8)
iKKMEC1	0,5000 (22)	0,4000 (22)	0,5600 (17)	0,4828 (14)	0,3323 (13)	0,1622 (19)
iKKMEC2	0,4583 (21)	0,3000 (18)	0,5600 (17)	0,4828 (14)	0,3323 (13)	0,0811 (8)
iKKMEC-GP1	0,4167 (14)	0,2000 (1)	0,4800 (15)	0,4483 (5)	0,3354 (20)	0,1351 (17)
iKKMEC-GP2	0,4167 (14)	0,2000 (1)	0,4400 (11)	0,4483 (5)	0,3354 (20)	0,1622 (19)
iKKMEC-GS1	0,3750 (5)	0,2000 (1)	0,3600 (1)	0,4138 (2)	0,3228 (11)	0,0811 (8)
iKKMEC-GS2	0,4167 (14)	0,2000 (1)	0,3600 (1)	0,4483 (5)	0,3228 (11)	0,1622 (19)
iKKMEC-LP1	0,5000 (22)	0,2000 (1)	0,4800 (15)	0,4138 (2)	0,3323 (13)	0,0541 (1)
iKKMEC-LP2	0,4167 (14)	0,2000 (1)	0,4400 (11)	0,4138 (2)	0,3323 (13)	0,0541 (1)
iKKMEC-LS1	0,4167 (14)	0,2000 (1)	0,3600 (1)	0,4828 (14)	0,2943 (1)	0,0541 (1)
iKKMEC-LS2	0,4167 (14)	0,2000 (1)	0,3600 (1)	0,4483 (5)	0,2943 (1)	0,1081 (16)

Tabela 5.11: Classificação de desempenho médio e mediano dos algoritmos de agrupamentos de acordo com o Índice Corrigido de Rand (CR) e a Taxa Total de Erro de Classificação (OERC).

	CR		OERC	
	D.médio	D.mediano	D.médio	D.mediano
iKM	17,9 (23)	19,5 (23)	15,3 (20)	17,0 (22)
iKKmeansEE	14,3 (18)	16,5 (17)	12,5 (18)	14,5 (18)
iKKmeansEC	17,3 (21)	17,5 (20)	15,8 (21)	16,5 (19)
iKKMEE1	13,0 (17)	16,5 (17)	11,6 (17)	12,0 (17)
iKKMEE2	16,3 (20)	18,0 (21)	13,3 (19)	16,5 (19)
iKKMEE-GP1	8,6 (12)	9,0 (11)	5,8 (9)	1,5 (4)
iKKMEE-GP2	8,0 (10)	7,0 (8)	7,6 (11)	5,0 (9)
iKKMEE-GS1	6,3 (5)	5,0 (4)	4,3 (4)	3,0 (5)
iKKMEE-GS2	5,5 (2)	4,0 (2)	3,2 (1)	1,0 (1)
iKKMEE-LP1	7,1 (7)	5,0 (4)	4,5 (5)	3,0 (5)
iKKMEE-LP2	7,7 (9)	10,0 (12)	5,1 (7)	3,5 (8)
iKKMEE-LS1	5,0 (1)	4,5 (3)	3,4 (3)	1,0 (1)
iKKMEE-LS2	5,5 (2)	3,5 (1)	3,3 (2)	1,0 (1)
iKKMEC1	17,8 (22)	18,5 (22)	17,5 (23)	17,5 (23)
iKKMEC2	16,2 (19)	17,0 (19)	15,8 (22)	16,5 (19)
iKKMEC-GP1	11,3 (16)	10,5 (14)	10,6 (16)	10,0 (13)
iKKMEC-GP2	11,2 (15)	12,5 (16)	10,1 (15)	10,0 (13)
iKKMEC-GS1	7,2 (8)	5,0 (4)	5,3 (8)	5,0 (9)
iKKMEC-GS2	8,6 (12)	7,0 (8)	7,6 (11)	6,5 (12)
iKKMEC-LP1	9,8 (14)	12,0 (15)	9,9 (14)	11,5 (16)
iKKMEC-LP2	8,4 (11)	10,0 (12)	8,0 (13)	10,5 (15)
iKKMEC-LS1	7,0 (6)	7,5 (10)	5,0 (6)	3,0 (5)
iKKMEC-LS2	6,0 (4)	5,5 (7)	5,8 (9)	5,0 (9)

Tabela 5.12: Média do desempenho médio e mediano segundo as funções *kernels* de uma (1 C) e duas (2 C) componentes.

	D.médio		D.mediano	
	1 C	2 C	1 C	2 C
iKKMEE	12,3 (17)	14,8 (19)	14,3 (17)	17,3 (21)
iKKMEE-GP	7,2 (10)	7,8 (11)	5,3 (7)	6 (10)
iKKMEE-GS	5,3 (4)	4,3 (2)	4,0 (4)	2,5 (2)
iKKMEE-LP	5,8 (5)	6,4 (9)	4,0 (4)	6,8 (11)
iKKMEE-LS	4,2 (1)	4,4 (3)	2,8 (3)	2,3 (1)
iKKMEC	17,7 (23)	16 (20)	18,0 (22)	16,8 (19)
iKKMEC-GP	10,9 (16)	10,6 (15)	10,3 (13)	11,3 (15)
iKKMEC-GS	6,2 (8)	8,1 (12)	5,0 (6)	6,8 (11)
iKKMEC-LP	9,9 (14)	8,2 (13)	11,8 (16)	10,3 (13)
iKKMEC-LS	6,0 (7)	5,9 (6)	5,3 (7)	5,3 (7)

Para grande maioria dos conjuntos de dados observou-se que pelo menos um algoritmo baseado em distâncias adaptativas apresentou desempenho superior aos métodos de agrupamento convencional. No pior das hipóteses, o desempenho foi equivalente. Olhando para o índice CR, em todos os casos, o melhor desempenho foi de um algoritmo baseado em distâncias adaptativas, enquanto que para o índice OERC, em todos os casos aconteceu o mesmo cenário, exceto nos conjuntos *Água* e *Temperatura*, onde o desempenho foi o mesmo.

Ao olhar para o desempenho médio e mediano de cada método de agrupamento, observou que os algoritmos com a restrição de que a soma dos pesos das variáveis deve ser igual a um (iKKMEE-GS1, iKKMEE-GS2, iKKMEE-LS1, iKKMEE-LS2, iKKMEC-GS1, iKKMEC-GS2, iKKMEC-LS1 e iKKMEE-LS2) foram superiores aos algoritmos com a restrição de que o produto dos pesos das variáveis deve ser igual a um (iKKMEE-GP1, iKKMEE-GP2, iKKMEE-LP1, iKKMEE-LP2, iKKMEC-GP1, iKKMEC-GP2, iKKMEC-LP1 e iKKMEE-LP2).

Observou-se também que, em geral, os algoritmos que consideraram que os protótipos eram definidos no espaço original dos dados (kernelização da métrica) apresentaram desempenho superior aos métodos que consideraram que os protótipos estavam definido no espaço de característica. Além disso, foi visto que os métodos baseado na distância não-adaptativa (iKKMEE1, iKKMEE2, iKKMEC1 e iKKMEC2) apresentaram desempenho inferiores aos métodos baseado em distâncias adaptativas. Ainda nesse ponto, tais métodos apresentaram desempenho médio e mediano semelhantes aos métodos convencionais.

Pode-se observar que a classificação de desempenho não apresentou diferenças segundo o tipo de função de *kernel* utilizado, de modo que em alguns casos os métodos que usavam o *kernel* de uma componente apresentaram desempenho superior, enquanto que em outros ocorreu o contrário. Em geral os melhores resultados foram obtidos pelos algoritmos iKKMEE-LS e iKKMEE-GS.

Por fim, olhando de modo geral para todos os algoritmos de agrupamento, os que apresentam melhores desempenho foram: iKKMEE-LS2, iKKMEE-GS2, iKKMEE-LS1 e iKKMEE-GS1, enquanto que, os algoritmos de agrupamento convencionais (iKM, iKKmeansEE e iKKmeansEC) e os algoritmos baseado na distância não-adaptativa (iKKMEE1, iKKMEE2, iKKMEC1, e iKKMEC2) apresentaram piores resultados.

CAPÍTULO 6

Conclusões

Nesse trabalho foi apresentado um conjunto de métodos de agrupamento *hard* para dados simbólicos do tipo intervalo com ponderação automática das variáveis através de distâncias adaptativas que eram obtidas como soma de distâncias euclidianas entre as observações e os centroides de cada grupo, calculadas individualmente para cada variável via funções *kernel*. Os métodos propostos apresentou um grande diferencial em relação aos métodos convencionais por causa da aprendizagem dos pesos das variáveis, que foi possibilitado através do uso de distâncias adaptativas, que podem ser a mesma para todos os grupos (distâncias adaptativas globais) e diferentes para cada grupo (distâncias adaptativas locais). O uso de funções *kernel* se justifica pelo fato de que em situações de disposição não-linear dos grupos, essas funções auxiliam a identifica-los. O *kernel* para intervalos utilizado nesse trabalho foi o *gaussiano* onde foi considerado duas versões: com uma e duas componentes. Para os algoritmos propostos foram obtidas as expressões para o cálculo dos protótipos, pesos e regra de alocação das observações que minimizavam o critério entre padrões e protótipos, como também foram demonstradas as propriedades de convergências dos algoritmos propostos.

Para os conjuntos de dados simulados, ficou claro a superioridade dos métodos de agrupamento baseados em distâncias adaptativas em relação aos métodos convencionais. No caso em que os grupos apresentavam um conjunto de variáveis importantes diferentes (conjunto de dados simulado 1), os algoritmos baseados em distâncias adaptativas

locais apresentaram melhor desempenho que os demais. No caso em que os grupos apresentam o mesmo conjunto de variáveis importantes (conjunto de dados simulados 2), os algoritmos baseados em distâncias globais apresentaram ótimo desempenho quando comparado com o cenário 1, apesar de que os métodos baseados em distâncias adaptativas locais apresentaram desempenho levemente superior. Os métodos de agrupamento baseados na distância não-adaptativa, mas com funções *kernel* separadas para cada variável (iKKMEE1, iKKMEE2, iKKMEC1 e iKKMEC2) não apresentam resultados satisfatórios quanto aos algoritmos baseados em distâncias adaptativas. Em muitos casos esses métodos apresentaram desempenho inferior aos métodos convencionais. Ainda na situação de dados sintéticos, através de testes não paramétricos, concluímos que os métodos baseados em distâncias adaptativas que usavam a função *kernel* de uma componente apresentaram, estatisticamente, desempenho igual ou superior quando comparado com sua versão que utilizava o *kernel* de duas componentes.

Com relação aos conjuntos de dados reais, mais uma vez, os algoritmos baseados em distâncias adaptativas apresentaram desempenho superior em relação aos métodos convencionais. Nessa ocasião, os algoritmos baseado em distâncias adaptativas com restrição de que a soma dos pesos das variáveis deve ser igual a um e sob a abordagem da kernelização da métrica (iKKMEE-GS1, iKKMEE-GS2, iKKMEE-LS1 e iKKMEE-LS2) apresentaram as melhores classificações de desempenho. De modo geral, os algoritmos sob a restrição de que a soma dos pesos das variáveis deve ser igual a um (iKKMEE-GS1, iKKMEE-GS2, iKKMEE-LS1, iKKMEE-LS2, iKKMEC-GS1, iKKMEC-GS2, iKKMEC-LS1 e iKKMEC-LS2) apresentaram melhores classificações de desempenho quando comparado com os métodos sob a restrição de que o produto dos pesos das variáveis deve ser igual a um (iKKMEE-GP1, iKKMEE-GP2, iKKMEE-LP1, iKKMEE-LP2, iKKMEC-GP1, iKKMEC-GP2, iKKMEC-LP1 e iKKMEC-LP2). Novamente, os métodos de agrupamento baseados na distância não-adaptativa, mas com funções *kernel* separadas para cada variável (iKKMEE1, iKKMEE2, iKKMEC1 e iKKMEC2) não apresentaram classificações de desempenho satisfatórias quanto aos algoritmos baseados em distâncias adaptativas. Em muitos casos esses métodos apresentaram piores classificações em relação aos métodos convencionais. Quando comparado os algoritmos segundo o uso da função *kernel* de uma e duas componentes, concluímos que para os dados reais, no geral, não houve uma superioridade por alguma das funções *kernel*. Em alguns casos os métodos que consideravam o *kernel* de uma componente apresentaram classificação de desempenho superior, enquanto em outros casos os métodos de agrupamento baseado no *kernel* de duas componentes se sobressaíram.

6.1 Trabalhos futuros

Alguns trabalhos podem ser desenvolvidos a partir dos temas abordados nesse trabalho. Dentre os possíveis assuntos podemos listar:

- Considerar diferentes funções *kernel* para dados do tipo intervalo (por exemplo, o *kernel* polinomial de grau d , tangente hiperbólico, Cauchy, etc.) e fazer todos os passos feitos nesse trabalho (encontrar as expressões para os protótipos, pesos, etc);
- Encontrar a expressão para o parâmetro σ^2 ótimo, e considerar que este se atualize a cada iteração do algoritmo;
- Estender o que foi feito nesse trabalho para a abordagem de agrupamento *fuzzy*;
- Disponibilizar as implementações no github e implementar um pacote para o R (R Core Team, 2013).

Qualquer dúvida, entre em contato através dos endereços:

1. José Nataniel - andradesa48@gmail.com
2. Marcelo Rodrigo - marcelorpf@gmail.com

Referências Bibliográficas

- ABRAHAM, A.; DAS, S.; KONAR, A. Document clustering using differential evolution. In: IEEE. *2006 IEEE International Conference on Evolutionary Computation*. [S.l.], 2006. p. 1784–1791.
- ARTHUR, D.; VASSILVITSKII, S. *k-means++: The advantages of careful seeding*. [S.l.], 2006.
- BEN-HUR, A. et al. Support vector clustering. *Journal of machine learning research*, v. 2, n. Dec, p. 125–137, 2001.
- BILLARD, L.; DIDAY, E. From the statistics of data to the statistics of knowledge: symbolic data analysis. *Journal of the American Statistical Association*, Taylor & Francis, v. 98, n. 462, p. 470–487, 2003.
- BOCK, H.; DIDAY, E. *Analysis of Symbolic Data, Exploratory methods for extracting statistical information from complex data*. [S.l.]: Springer, 2000.
- BREIMAN, L. et al. Classification and regression trees chapman & hall. *New York*, 1984.
- CAPUTO, B. et al. Appearance-based object recognition using svms: which kernel should i use? 2001.
- CARVALHO, F. d. A. de; BRITO, P.; BOCK, H.-H. Dynamic clustering for interval data based on l 2 distance. *Computational Statistics*, Springer, v. 21, n. 2, p. 231–250, 2006.
- CHEN, D.; ZHANG, S. Fuzzy clustering using kernel method. *IEEE, Nanjing*, 2002.
- CHIANG, J.-H.; HAO, P.-Y. A new kernel-based fuzzy clustering approach: support vector clustering with cell growing. *IEEE Transactions on Fuzzy Systems*, IEEE, v. 11, n. 4, p. 518–527, 2003.
- CHOUDHARI, A. et al. Mesh based clustering without stopping criterion. In: IEEE. *2005 Annual IEEE India Conference-Indicon*. [S.l.], 2005. p. 530–534.

- COSTA, A. F.; PIMENTEL, B. A.; SOUZA, R. M. de. A kernel k-means clustering method for symbolic interval data. In: IEEE. *The 2010 International Joint Conference on Neural Networks (IJCNN)*. [S.l.], 2010. p. 1–6.
- COSTA, A. F. B. F. d. Agrupamento de dados simbólicos intervalares usando funções de kernel. Universidade Federal de Pernambuco, 2011.
- CUI, X.; POTOK, T. E. Document clustering analysis based on hybrid pso+ k-means algorithm. *Journal of Computer Sciences (special issue)*, Citeseer, v. 27, p. 33, 2005.
- DIDAY, E. *Eléments d'analyse de données*. [S.l.]: Bordas Editions, 1982.
- DIDAY, E. *Orders and overlapping clusters in pyramids*. In de Leew, J. et al.(eds) *Multidimensional Data Analysis*. [S.l.]: DSWO Press, Leiden, 1986.
- DIDAY, E.; SIMON, J. *Digital Pattern Classification*. [S.l.]: Springer, Berlin, 1976.
- FERREIRA, M. R. P. Agrupamento baseado em kernel com ponderação automática das variáveis via distâncias adaptativas. Universidade Federal de Pernambuco, 2013.
- FILIPPONE, M. et al. A survey of kernel and spectral methods for clustering. *Pattern recognition*, Elsevier, v. 41, n. 1, p. 176–190, 2008.
- GIROLAMI, M. Mercer kernel-based clustering in feature space. *IEEE Transactions on Neural Networks*, Citeseer, v. 13, n. 3, p. 780–784, 2002.
- GRAEPEL, T.; OBERMAYER, K. Fuzzy topographic kernel clustering. In: *Proceedings of the 5th GI Workshop Fuzzy Neuro Systems*. [S.l.: s.n.], 1998. v. 98, p. 90–97.
- GURU, D.; KIRANAGI, B. B.; NAGABHUSHAN, P. Multivalued type proximity measure and concept of mutual similarity value useful for clustering symbolic patterns. *Pattern Recognition Letters*, Elsevier, v. 25, n. 10, p. 1203–1213, 2004.
- HAYES-ROTH, F.; MCDERMOTT, J. An interference matching technique for inducing abstractions. *Communications of the ACM*, ACM, v. 21, n. 5, p. 401–411, 1978.
- HAYKIN, S. *Neural networks: a comprehensive foundation*. [S.l.]: Prentice Hall PTR, 1994.
- HÖPPNER, F. et al. *Fuzzy cluster analysis: methods for classification, data analysis and image recognition*. [S.l.]: John Wiley & Sons, 1999.
- HUBERT, L.; ARABIE, P. Comparing partitions. *Journal of classification*, Springer, v. 2, n. 1, p. 193–218, 1985.
- JAIN, A. K. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, Elsevier, v. 31, n. 8, p. 651–666, 2010.
- KIM, D.-W. et al. A kernel-based subtractive clustering method. *Pattern Recognition Letters*, Elsevier, v. 26, n. 7, p. 879–891, 2005.
- MACQUEEN, J. et al. Some methods for classification and analysis of multivariate observations. In: OAKLAND, CA, USA. *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. [S.l.], 1967. v. 1, n. 14, p. 281–297.

- MERCER, B. J. Functions of positive and negative type, and their connection the theory of integral equations. *Phil. Trans. R. Soc. Lond. A*, The Royal Society, v. 209, n. 441-458, p. 415–446, 1909.
- MICHALSKI, R. S. Aqval/1—computer implementation of a variable-valued logic system vl1 and examples of its application to pattern recognition. 1973.
- MILLIGAN, G. W.; COOPER, M. C. A study of the comparability of external criteria for hierarchical cluster analysis. *Multivariate behavioral research*, Taylor & Francis, v. 21, n. 4, p. 441–458, 1986.
- MULLER, K.-R. et al. An introduction to kernel-based learning algorithms. *IEEE transactions on neural networks*, IEEE, v. 12, n. 2, p. 181–201, 2001.
- MURTY, M. N.; JAIN, A.; FLYNN, P. J. Data clustering: a review. *ACM Computing Surveys*, v. 31, n. 3, p. 264–323, 1999.
- OLIVEIRA, C. R. d. et al. Agrupamento subtrativo baseado em kernel para dados simbólicos da natureza intervalar. Universidade Federal da Paraíba, 2018.
- R Core Team. *R: A Language and Environment for Statistical Computing*. Vienna, Austria, 2013. Disponível em: <<http://www.R-project.org/>>.
- RUSSELL, S. J.; NORVIG, P. *Artificial intelligence: a modern approach*. [S.l.]: Malaysia; Pearson Education Limited,, 2016.
- SCHÖLKOPF, B.; SMOLA, A.; MÜLLER, K.-R. Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, MIT Press, v. 10, n. 5, p. 1299–1319, 1998.
- SCHÖLKOPF, B. et al. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. [S.l.]: MIT press, 2002.
- SNEATH, P. H.; SOKAL, R. R. et al. *Numerical taxonomy. The principles and practice of numerical classification*. [S.l.: s.n.], 1973.
- STEINHAUS, H. Sur la division des corp materiels en parties. *Bull. Acad. Polon. Sci*, v. 1, n. 804, p. 801, 1956.
- WARWICK, K. M.; MORINEAU, A. *Multivariate descriptive statistical analysis*. [S.l.]: John Wiley & Sons, Inc., 1984.