Agrupamento de fluxos de dados utilizando dimensão fractal

Christian Cesar Bones

Tese de Doutorado do Programa de Pós-Graduação em Ciências de
Computação e Matemática Computacional (PPG-CCMC)

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Christian Cesar Bones

Agrupamento de fluxos de dados utilizando dimensão fractal

Tese apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Doutor em Ciências – Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientadora: Profa. Dra. Elaine Parros Machado de Sousa

USP – São Carlos
Maio de 2018

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

B712a Bones, Christian Cesar
 Agrupamento de fluxos de dados utilizando
 dimensão fractal / Christian Cesar Bones;
 orientadora Elaine Parros Machado de Sousa. -- São
 Carlos, 2018.
 140 p.

 Tese (Doutorado - Programa de Pós-Graduação em
 Ciências de Computação e Matemática Computacional) --
 Instituto de Ciências Matemáticas e de Computação,
 Universidade de São Paulo, 2018.

 1. Mineração de dados. 2. Fluxo de dados. 3.
 Agrupamento. 4. Sensores. I. Sousa, Elaine Parros
 Machado de, orient. II. Título.

Christian Cesar Bones

Clustering data streams using fractal dimension

Doctoral dissertation submitted to the Institute of Mathematics and Computer Sciences – ICMC-USP, in partial fulfillment of the requirements for the degree of the Doctorate Program in Computer Science and Computational Mathematics. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Profa. Dra. Elaine Parros Machado de Sousa

USP – São Carlos
May 2018

*Este trabalho é dedicado à minha esposa,
família e amigos. Amo vocês!*

AGRADECIMENTOS

A *Deus*, por conceder essa oportunidade de aprendizado e por ter colocado em meu caminho pessoas tão amadas.

A minha esposa *Clelia Junko Kinzu Dimário*, por todo o amor, carinho, apoio, incentivo e compreensão, sem os quais eu não estaria aqui hoje. Por você ser essa pessoa maravilhosa, adorável e íntegra. Te amo muito!

A meus Pais, *Ademar Solano Bones* e *Eli Maria Bones*, por todos os valiosos ensinamentos, amor, carinho e incentivo.

A Professora Dra *Elaine Parros Machado de Sousa* por sua orientação primorosa, pelos valiosos conselhos, disponibilidade, empenho e dedicação dirigidos a mim.

Aos meus irmãos, *Márcio Rodrigo Bones* e *Paulo Henrique Bones*, pelo apoio e carinho de sempre.

Ao meu novo irmão, *Luiz Olmes Carvalho* provando que laços eternos também são constituídos pela amizade e esquisitices.

Ao meu grande amigo/afiliado *Lúcio Fernandes Dutra Santos*, pelos incontáveis momentos de alegria, apoio e auxílio.

Ao meu grande amigo *Willian Dener de Oliveira*, pelo companheirismo e amizade.

A CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) e a FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo) pelos suportes fornecidos durante esse doutoramento.

A EMBRAPA (Empresa Brasileira de Pesquisa Agropecuária) e seus pesquisadores pelo apoio e fornecimento do conjunto de dados utilizado.

Aos *professores e funcionários do Instituto de Ciências Matemáticas e de Computação*, pelo conhecimento compartilhado e profissionalismo, e à *Universidade de São Paulo*, pelo uso de suas instalações, recursos e equipamentos.

A todos os amigos que cruzaram o meu caminho e deixaram marcas e saudades.

*“Deus nos concede, a cada dia, uma página de vida nova no livro do tempo.
Aquilo que colocarmos nela, corre por nossa conta.”
(Francisco Cândido Xavier)*

RESUMO

BONES, C. C. **Agrupamento de fluxos de dados utilizando dimensão fractal**. 2018. 140 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2018.

Realizar o agrupamento de fluxos de dados contínuos e multidimensionais (*multidimensional data streams*) é uma tarefa dispendiosa, visto que esses tipos de dados podem possuir características peculiares e que precisam ser consideradas, dentre as quais destacam-se: podem ser infinitos, tornando inviável, em muitas aplicações realizar mais de uma leitura dos dados; ponto de dados podem possuir diversas dimensões e a correlação entre as dimensões pode impactar no resultado final da análise e; são capazes de evoluir com o passar do tempo. Portanto, faz-se necessário o desenvolvimento de métodos computacionais adequados a essas características, principalmente nas aplicações em que realizar manualmente tal tarefa seja algo impraticável em razão do volume de dados, por exemplo, na análise e predição do comportamento climático. Nesse contexto, o objetivo desse trabalho de pesquisa foi propor técnicas computacionais, eficientes e eficazes, que contribuíssem para a extração de conhecimento de fluxos de dados com foco na tarefa de agrupamento de fluxos de dados similares. Assim, no escopo deste trabalho, foram desenvolvidos dois métodos para agrupamento de fluxos de dados evolutivos, multidimensionais e potencialmente infinitos, ambos baseados no conceito de dimensão fractal, até então não utilizada nesse contexto na literatura: o eFCDS, acrônimo para *evolving Fractal Clustering of Data Streams*, e o eFCC, acrônimo para *evolving Fractal Clusters Construction*. O eFCDS utiliza a dimensão fractal para mensurar a correlação, linear ou não, existente entre as dimensões dos dados de um fluxo de dados multidimensional num período de tempo. Esta medida, calculada para cada fluxo de dados, é utilizada como critério de agrupamento de fluxos de dados com comportamentos similares ao longo do tempo. O eFCC, por outro lado, realiza o agrupamento de fluxos de dados multidimensionais de acordo com dois critérios principais: comportamento ao longo do tempo, considerando a medida de correlação entre as dimensões dos dados de cada fluxo de dados, e a distribuição de dados em cada grupo criado, analisada por meio da dimensão fractal do mesmo. Ambos os métodos possibilitam ainda a identificação de *outliers* e constroem incrementalmente os grupos ao longo do tempo. Além disso, as soluções propostas para tratamento de correlações em fluxos de dados multidimensionais diferem dos métodos apresentados na literatura da área, que em geral utilizam técnicas de sumarização e identificação de correlações lineares aplicadas apenas à fluxos de dados unidimensionais. O eFCDS e o eFCC foram testados e confrontados com métodos da literatura que também se propõem a agrupar fluxos de dados. Nos experimentos realizados com dados sintéticos e reais, tanto o eFCDS quanto o eFCC obtiveram maior eficiência na construção dos agrupamentos, identificando os fluxos de dados com comportamento semelhante e cujas dimensões se correlacionam de maneira similar.

Além disso, o eFCC conseguiu agrupar os fluxos de dados que mantiveram distribuição dos dados semelhante em um período de tempo. Os métodos possuem como uma das aplicações imediatas a extração de padrões de interesse de fluxos de dados proveniente de sensores climáticos, com o objetivo de apoiar pesquisas em Agrometeorologia.

Palavras-chave: Mineração de dados, Fluxo de dados, Agrupamento, Sensores.

ABSTRACT

BONES, C. C. **Clustering data streams using fractal dimension**. 2018. 140 p. Thesis (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2018.

To cluster multidimensional data streams is an expensive task since this kind of data could have some peculiarities characteristics that must be considered, among which: they are potentially infinite, making many reads impossible to perform; data can have many dimensions and the correlation among them could have an affect on the analysis; as the time pass through they are capable of evolving. Therefore, it is necessary the development of appropriate computational methods to these characteristics, especially in the areas where performing such task manually is impractical due to the volume of data, for example, in the analysis and prediction of climate behavior. In that context, the research goal was to propose efficient and effective techniques that clusters multidimensional evolving data streams. Among the applications that handles with that task, we highlight the *evolving Fractal Clustering of Data Streams*, and the eFCC acronym for *evolving Fractal Clusters Construction*. The eFCDS calculates the data streams' fractal dimension to correlate the dimensions in a non-linear way and to cluster those with the biggest similarity over a period of time, evolving the clusters as new data is read. Through calculating the fractal dimension and then cluster the data streams the eFCDS applies an innovative strategy, distinguishing itself from the state-of-art methods that perform clustering using summaries techniques and linear correlation to build their clusters over unidimensional data streams. The eFCDS also identifies those data streams who showed anomalous behavior in the analyzed time period treating them as outliers. The other method devoleped is called eFCC. It also builds data streams clusters, however, they are built on a two premises basis: the data distribution should be likely the same and second the behavior should be similar in the same time period. To perform that kind of clustering the eFCC calculates the clusters' fractal dimension itself and the data streams' fractal dimension, following the evolution in the data, relocating the data streams from one group to another when necessary and identifying those that become outlier. Both eFCDS and eFCC were evaluated and confronted with their competitor, that also propose to cluster data streams and not only data points. Through a detailed experimental evaluation using synthetic and real data, both methods have achieved better efficiency on building the groups, better identifying data streams with similar behavior during a period of time and whose dimensions correlated in a similar way, as can be observed in the result chapter 6. Besides that, the eFCC also cluster the data streams which maintained similar data distribution over a period of time. As immediate application the methods developed in this thesis can be used to extract patterns of interest from climate sensors aiming to support researches in agrometeorology.

Keywords: Data Mining, Data Streams, Clustering, Sensors .

LISTA DE ILUSTRAÇÕES

Figura 1 – Fluxo de dados multidimensional: radiação solar, temperatura máxima e mínima, gerado pelo sensor agrometeorológico de Sete Lagoas - MG no ano de 1960.	31
Figura 2 – Processo de extração de conhecimento, adaptado de (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996).	32
Figura 3 – Fontes geradoras de fluxos de dados enviando dados para entidade de processamento.	40
Figura 4 – Exemplo de janela deslizante de quantidade fixa de dados	43
Figura 5 – Exemplo de janela deslizante com início fixado no tempo - <i>Landmark Windows</i>	44
Figura 6 – Exemplo de histogramas: (a) simétrico, com o pico localizado no centro do gráfico; (b) assimétrico, com o pico deslocado do centro	46
Figura 7 – Histograma width, com <i>buckets</i> construído com base nos limites inferior e superior do vetor $V = 1, 2; 1, 4; 2, 1; 3, 4; 3, 5; 3, 4; 2, 3; 4, 3; 4, 4; 4, 2; 4, 4; 4, 5; 1, 5$	47
Figura 8 – Exemplo da aplicação da transformada de Haar ao vetor A	48
Figura 9 – Exemplo construção e identificação do caminho mínimo na Matriz gerado pela técnica DTW.	50
Figura 10 – Representação gráfica do alinhamento obtido pela DTW de dois fluxos de dados (A,B).	50
Figura 11 – Exemplo de agrupamento de ponto de dados com resultado inconsistente no grupo B.	55
Figura 12 – Hierarquia de grupos, onde os pontos de dados provenientes do fluxo de dados B são mais similares aos pontos de dados do fluxo de dados C, do mesmo modo E é mais similar a F e G possui similaridade acentuada com H, com o refinamento da hierarquia ABC e EFGH são agrupados, posteriormente D é agrupado a ABC e finalmente a hierarquia é completada gerando o grupo ABDCEFGH.	57
Figura 13 – Diferença do agrupamento de dados para o agrupamento de fluxos de dados.	62
Figura 14 – Representação dos conceitos empregados em índices externos por meio de uma Matriz de Confusão.	75
Figura 15 – Fractais encontrados na natureza	78
Figura 16 – Fractais definidos matematicamente	79
Figura 17 – Reta em espaços de dimensões 2 e 3, onde $\mathcal{D}_2 = 1$	81

Figura 18 – Representação da Árvore de Contagem	82
Figura 19 – Representação de um hipercubo bidimensional com três níveis de estrutura e da Árvore de Contagem Continuada.	84
Figura 20 – Fluxo de dados de 3 dimensões com os eventos $e_i = (a_1, a_2, a_3)$ recebidos em períodos de contagem de uma janela móvel.	85
Figura 21 – Exemplo ilustrado de agrupamento de fluxos de dados, no qual cada SX_{FD} representa um sensor que é agrupado usando a medida de dimensão fractal $\mathcal{D}_2$ no tempo T_i	92
Figura 22 – Visão geral do método eFCDS para agrupar fluxos de dados.	92
Figura 23 – A janela deslizante (W) de um sensor, períodos de contagem $\eta_c = 4$, eventos $e = 3$. No instante T1, a janela inicia no período 1 e termina no período 4; depois deslocando a janela para T2, (W) inicia no período 2 e termina no período 5.	93
Figura 24 – Exemplo de redução de dados transferido para o módulo de mineração pelo módulo de análise de dimensão fractal. Em cada passo da janela deslizante (t_i) é transferida apenas uma dimensão fractal para o módulo de mineração.	94
Figura 25 – (a) Sobreposição entre os grupos; (b) Primeiro tenta realizar a união dos grupos sem extrapolar o máximo desvio padrão σ_{MAX} ; (c) Se falhar na união, então um novo grupo é criado e é testado se os fluxos de dados permanecerão no seu grupo original ou migrarão para o novo grupo.	96
Figura 26 – Processo para verificar se um fluxo de dados pode ser incluído em um grupo baseando-se na expansão do mesmo.	98
Figura 27 – Visão geral do eFCC para agrupar fluxos de dados.	99
Figura 28 – Processo para verificar os fluxos de dados que estão em dissonância com o grupo que estão inseridos.	103
Figura 29 – Conjuntos sintéticos	111
Figura 30 – Conjunto de fluxos de dados sintéticos gerados pelas Equações 6.1 até 6.4.	112
Figura 31 – Evolução dos grupos formados em cada uma das iterações.	118
Figura 32 – Resultados comparativos das porcentagens de desvio padrão dos grupos formados pelos métodos	118
Figura 33 – Resultado de Silhueta do eFCDS.	119
Figura 34 – Resultado de Silhueta do eFCC.	120
Figura 35 – Resultado da Silhueta obtida pelo ECM para os agrupamentos da primeira e última iteração	121
Figura 36 – Resultados do cálculo da Silhueta dos agrupamentos obtidos pelo POD-Clus para cada uma das dimensões dos fluxos de dados	122
Figura 37 – Resultado comparativo da distribuição dos dados nos grupos entre o eFCDS e eFCC, grupo de Juíz de Fora - MG (JDFO).	123

Figura 38 – Resultado comparativo da distribuição dos dados nos grupos entre o eFCDS e eFCC, comparando outras estações do grupo de Juíz de Fora - MG (JDFO).	123
Figura 39 – Tempo gasto de processamento em minutos para cada um dos métodos para processar todo o conjunto de dados (21024000 medidas).	124

LISTA DE QUADROS

Quadro 1 – Quadro comparativo entre os métodos apresentados	76
Quadro 2 – Sumarização das características dos conjuntos sintéticos	110
Quadro 3 – Resultado obtido pelo ECM variando o parâmetro de limiar	113
Quadro 4 – Resultado obtido pelo PODClus variando o parâmetros de limiar de similaridade.	114
Quadro 5 – Resultado obtido pelo eFCDS variando o parâmetro de máximo desvio padrão (σ_{max})	115
Quadro 6 – Resultado obtido pelo eFCC variando os parâmetros máximo desvio padrão (σ_{max}) e limiar de expansão ou contração τ	116
Quadro 7 – Sumarização dos resultados obtidos pelos métodos, para cada uma das situações apresentadas	116

LISTA DE ALGORITMOS

Algoritmo 1 – Algoritmo que testa a divisão de dois grupos	66
Algoritmo 2 – Algoritmo que testa a agregação de dois grupos	66
Algoritmo 3 – Algoritmo ODAC - Online Divisive Agglomerative Clustering	67
Algoritmo 4 – ECM	69
Algoritmo 5 – <i>Linear Box Occupancy Counter</i> - LiBOC	83
Algoritmo 6 – Agrupamento evolutivo baseado em fractal de <i>Data Streams</i> (eFCDS) .	94
Algoritmo 7 – Função para unir grupos em sobreposição	97
Algoritmo 8 – Busca e Inserção de fluxo de dados em um Grupo - BIG	101
Algoritmo 9 – Atualiza o grupo atual do <i>Data Stream</i>	101
Algoritmo 10 – Agrupamento evolutivo de <i>Data Streams</i> (eFCC)	102
Algoritmo 11 – Busca e reagrupa <i>Outliers</i> - BRO	104

LISTA DE ABREVIATURAS E SIGLAS

ADWIN	ADaptive sliding WINdow
ARI	Adjusted Rand Index
CF	Clustering Features
DBMS	Database Management System
DE	Distância Euclidiana
DF	Dimensão Fractal
DSMS	Data Stream Management Systems
DTW	Dynamic Time Warping
ECM	Evolving Clustering Method
eFCC	evolving Fractal Clusters Construction
eFCDS	evolving Fractal Clustering of Data Streams
HWT	Haar Wavelet Transform
KDD	Knowledge Discovery Database
LCS	Longest Common Subsequence
ODAC	Online Divisive Agglomerative Clustering
PODClus	Probability and Distribution-based Clustering
RNOMC	Rooted Normalized One-Minus-Correlation
SSE	Soma dos Erros ao Quadrados - (Sum Square Error)
SSQ	Soma do Quadrado das Distâncias

LISTA DE SÍMBOLOS

x_i — Ponto de dado unidimensional.

S_i — *Data stream* qualquer pertencente a $\mathbb{S}$

x_i^d — Ponto de dados D-dimensional no instante de tempo i .

$\mathbb{S}$ — conjunto de fluxos de dados, tal que, $\mathbb{S} = S_1, S_2, \dots, S_N$

$\mathbb{C}_{\square}$ — grupo contendo fluxos de dados multidimensionais (DS)

$\mathbb{P}$ — coleção de grupos disjuntos de C_i

$\mathbb{S}^D$ — conjunto de fluxos de dados multidimensional, tal que, $\mathbb{S}^D = DS_1, DS_2, \dots, DS_N$

E — dimensão de imersão

$\mathcal{D}$ — Dimensão Fractal

$\mathbb{L}$ — reticulado hiper-cúbico

r — tamanho do lado do reticulado hiper-cúbico

$\mathcal{D}_2$ — Dimensão de Correlação Fractal

e — eventos

η_c — número de períodos de contagem

η_e — número de eventos por período de contagem

R — quantidade de pontos no gráfico *box-counting plot*

SUMÁRIO

1	INTRODUÇÃO	29
1.1	Contexto e Motivação	30
1.2	Definição do Problema	32
1.3	Lacuna de Pesquisa e Objetivos do Trabalho	34
1.4	Principais Contribuições	36
1.5	Organização da Tese	37
2	FLUXOS DE DADOS	39
2.1	Conceitos de Fluxo de Dados	39
2.2	Delimitação de Fluxo de Dados	42
2.3	Sumarização de fluxos de dados	44
2.3.1	<i>Histograma</i>	45
2.3.2	<i>Wavelets</i>	46
2.3.3	<i>Funções de Distância</i>	48
2.4	Considerações finais	51
3	TAREFAS DE AGRUPAMENTO EM MINERAÇÃO DE FLUXOS DE DADOS	53
3.1	Técnicas de Agrupamento Aplicadas a Fluxos de Dados	54
3.1.1	<i>Agrupamentos Hierárquicos</i>	56
3.1.2	<i>Agrupamento baseado em particionamento</i>	58
3.1.2.1	<i>STREAM</i>	59
3.1.2.2	<i>CluStream</i>	59
3.1.3	<i>Agrupamento baseado em grade</i>	60
3.1.3.1	<i>FC</i>	61
3.1.4	<i>Agrupamento de Fluxo de Dados</i>	62
3.1.4.1	<i>Método de Beringuer e Hüllermeier</i>	63
3.1.4.2	<i>Online Divisive Agglomerative Clustering</i>	64
3.1.4.3	<i>POD-Clus</i>	67
3.1.4.4	<i>ECM</i>	68
3.1.4.5	<i>TS-Stream</i>	70
3.2	Medidas de Qualidade de Agrupamentos	71
3.2.1	<i>Índices Internos</i>	72

3.2.1.1	<i>Índice SSE</i>	72
3.2.1.2	<i>Coeficiente de Silhueta</i>	73
3.2.1.3	<i>Índice de Dunn</i>	73
3.2.2	<i>Índices Externos</i>	74
3.2.2.1	<i>Terminologia empregada em índices externos</i>	74
3.2.2.2	<i>Índice Precisão</i>	75
3.2.2.3	<i>Índice Revocação</i>	75
3.3	Considerações finais	75
4	TEORIA DE FRACTAIS	77
4.1	Definição	77
4.2	Dimensão Fractal	79
4.3	Dimensão Fractal de Fluxos de Dados	82
4.4	Considerações finais	86
5	AGRUPAMENTO DE FLUXOS DE DADOS E DETECÇÃO DE OUTLIERS BASEADO EM FRACTAIS	89
5.1	eFCDS	91
5.2	eFCC	97
5.3	Considerações Finais	105
6	EXPERIMENTOS	107
6.1	Dados Sintéticos	109
6.2	Dados Reais	117
6.3	Considerações finais	125
7	CONCLUSÃO	127
7.1	Principais Contribuições	128
7.2	Publicações	129
7.3	Propostas para Trabalhos Futuros	130
	REFERÊNCIAS	131

INTRODUÇÃO

Nos dias atuais, é crescente o número de dispositivos e sensores que geram dados continuamente. Por exemplo, as casas podem ser equipadas com sensores para as mais diversas finalidades, que incluem desde atividades simples como manter e monitorar a temperatura ambiente, até atividades complexas, como monitorar por meio de câmeras a saúde dos idosos ou a segurança da casa. Os sensores são também utilizados na área de engenharia elétrica, fornecendo medidas de consumo de energia para sistemas especialistas. Esses sistemas buscam prever a demanda por eletricidade em determinada região ou país, evitando que a produção seja menor que o consumo. Há ainda, os sensores de estações meteorológicas, que medem temperatura mínima e máxima, precipitação pluviométrica, entre outras medidas utilizadas na área de agrometeorologia. Nessa área, os sensores realizam a coleta das medidas e as transmitem para sistemas que procuram identificar a região que determinada cultura agrícola se desenvolverá melhor, prever a melhor época para realizar o plantio e a colheita avaliar os efeitos de eventos climáticos extremos ou de mudanças climáticas, dentre outros.

Os dados gerados por sensores vêm influenciando as relações e hábitos individuais tanto de forma econômica quanto social. Portanto, propor modelos computacionais precisos, eficientes e eficazes, baseados na análise e extração de conhecimento de dados de sensores, pode ser de grande valia para o auxílio à população nos mais diversos setores. Além disso, esses modelos podem auxiliar a pesquisa científica nas áreas em que o volume de dados torna proibitiva a análise apenas por especialistas, por exemplo, predição do comportamento climático, prevenção de desastres naturais e monitoramento de consumo de energia elétrica. No entanto, realizar a leitura, análise e a extração de conhecimento do grande fluxo de dados de uma rede de sensores é uma tarefa custosa e que demanda métodos computacionais sofisticados (GAMA, 2010), por se tratar de uma grande quantidade de sensores, produzindo dados de maneira contínua em um ambiente dinâmico (GAMA, 2012). Logo, desafios científicos inerentes a esses ambientes têm motivado diversos trabalhos de pesquisa envolvendo redes de sensores e mineração de fluxos de dados (*data stream mining*)(HYDE; ANGELOV; MACKENZIE, 2017; SILVA; HRUSCHKA;

GAMA, 2017; DING *et al.*, 2016; KHAN; HUANG; IVANOV, 2016; PEREIRA; MELLO, 2014; ??; FORESTIERO; PIZZUTI; SPEZZANO, 2013). E é nesse contexto que se insere esta tese de doutoramento.

1.1 Contexto e Motivação

O fluxo contínuo de dados gerados por diversas fontes, dentre as quais os sensores, é chamado de fluxo de dados. Conforme definido em (GOLAB, 2009), um fluxo de dados S é uma coleção ordenada de itens de dados $[x_1, x_2, \dots]$, gerados por uma ou mais fontes, sendo enviados para uma ou mais entidades de processamento. Entretanto, a ordem de chegada dos itens de dados em geral não é controlada pelas entidades de processamento (GOLAB, 2009). Outra definição para o modelo de fluxo de dados é fornecida por Guha *et al.* (2003), como sendo uma sequência ordenada de pontos que devem ser acessados em ordem e podendo ser lidos apenas uma vez ou uma quantidade limitada de vezes. Na literatura, os itens de dados, ou pontos de dados, de um fluxo de dados também podem ser chamados de eventos ou item. Neste trabalho, iremos utilizar o termo pontos de dados. Alguns exemplos de fluxos de dados podem ser: o tráfego das redes de computadores, buscas na web, medições de sensores de uma indústria petroquímica para aferir a quantidade de gases poluentes despejadas no meio ambiente ou dados coletados por estações meteorológicas.

Outra característica, presente em um fluxo de dados, é a dimensionalidade. Para conceitualizar a dimensionalidade de fluxo de dados, vamos utilizar como base a definição de Golab (2009) apresentada anteriormente. Então, seja S uma coleção ordenada de pontos de dados gerado por uma ou mais fontes de dados, para cada ponto de dados em um dado instante de tempo i é definido um espaço de dimensão d cuja quantidades de dimensões são finitas ($1 \leq d < \infty$) que representam os atributos existentes no fluxo de dados. Assim, é chamado de unidimensional o fluxo de dados que possui apenas um atributo para cada ponto de dado observado. Como exemplo de fluxos de dados unidimensionais temos os fluxos de dados gerados por sensores agrometeorológicos que coletam apenas informação da radiação solar. Por sua vez, um fluxo de dados multidimensional possui mais de um atributo para cada ponto de dados observado, por exemplo, fluxos de dados gerados por sensores que além da radiação solar, colem também a temperatura máxima e mínima. Desse modo, um fluxo de dados que colete tais tipos de dados possuirá três dimensões. A Figura 1 apresenta os dados de três dimensões (radiação Solar, temperatura máxima e temperatura mínima) geradas por um sensor agrometeorológico¹.

Como mencionado anteriormente, uma grande quantidade de dados multidimensionais pode ser gerada a cada milésimo de segundo em diversas fontes, tornando restritivos a realização de múltiplas leituras e o armazenamento de fluxos de dados multidimensionais. Nesse contexto, os desafios computacionais relacionados ao processamento e à extração de conhecimento aumentam

¹ Dados reais obtidos a partir do sensor agrometeorológico de Sete Lagoas - MG no ano de 1960.

Figura 1 – Fluxo de dados multidimensional: radiação solar, temperatura máxima e mínima, gerado pelo sensor agrometeorológico de Sete Lagoas - MG no ano de 1960.

(a) Representação 2D.

(b) Representação 3D.

Fonte: Elaborada pelo autor.

significativamente, dentre os quais pode-se citar ([AGGARWAL *et al.*, 2006](#); [GAMA, 2010](#); [AGGARWAL, 2013](#)):

- O grande aumento do volume de dados torna inviável processar fluxos de dados eficientemente usando diversas leituras sobre esses mesmos dados. O cenário mais factível é processar cada conjunto de dados uma única vez. Essa característica restringe a implementação dos algoritmos convencionais e, portanto, algoritmos de mineração de fluxos de dados precisam ser projetados e desenvolvidos para realizar, preferencialmente, apenas uma leitura dos dados.
- Com o passar do tempo os dados podem evoluir de diversas maneiras, por exemplo, mediante mudanças de comportamento, alteração na distribuição dos dados ou mudanças de padrões. Esse comportamento é chamado de localidade temporal em fluxo de dados. Então, apenas modificar os algoritmos para que somente uma leitura dos dados seja realizada não é garantia de resultados satisfatórios e eficientes. Os algoritmos de mineração de fluxo de dados devem, também, acompanhar a característica evolutiva dos dados.
- Apesar do fluxo contínuo e potencialmente infinito de dados, os algoritmos de fluxos de dados devem ser capazes de tratar esses fluxos utilizando recursos computacionais finitos, como poder de processamento, memória, comunicação e tempo de processamento.
- Com o advento dos dispositivos inteligentes, que buscam facilitar a vida dos usuários em tempo real, as respostas do processamento dos dados deverão estar disponíveis a qualquer momento.
- A coleta e o processamento dos dados poderão ser distribuídos.

Assim, trabalhar com fluxos de dados que podem evoluir e crescer ilimitadamente não é uma tarefa trivial. Realizar a mineração de dados e a extração de conhecimento de fluxos de dados pode ser ainda mais custoso, devido, principalmente, à restrição de apenas uma leitura do fluxo de dados e à característica evolutiva dos dados (AGGARWAL *et al.*, 2006). Nesse contexto, os desafios inerentes ao processamento e à extração de conhecimento de fluxos de dados têm motivado o desenvolvimento de diversos trabalhos de pesquisa em Banco de Dados e Mineração de Dados, visando apresentar soluções eficientes e eficazes para problemas em diversos domínios de aplicação.

1.2 Definição do Problema

A extração de conhecimento a partir de bases de dados envolve, principalmente, a utilização de métodos de análise de dados e algoritmos de mineração de dados para descoberta de padrões que, sob aceitáveis limitações de eficiência computacional, geram uma quantidade particular de padrões de interesse sobre os dados (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996). Segundo (GAMA, 2010) a extração de conhecimento de fluxos de dados não difere muito da extração de conhecimento de bases de dados, tendo que levar em consideração, principalmente, apenas uma leitura dos dados e o uso limitado de recursos computacionais. O processo de extração de conhecimento possui as etapas de seleção dos dados para obtenção dos dados alvo, pré-processamento dos dados selecionados, transformação dos dados, aplicação de métodos de mineração de dados para a busca de padrões nos dados e, por último, a avaliação dos padrões encontrados para gerar conhecimento. Esse processo de extração de conhecimento pode ser observado na Figura 2.

Figura 2 – Processo de extração de conhecimento, adaptado de (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996).

Fonte: Elaborada pelo autor.

No processo de extração de conhecimento de fluxos de dados, três tarefas são de fundamental importância: a delimitação (pré-processamento), a sumarização (transformação) e a mineração dos dados. A etapa de delimitação busca restringir a quantidade de dados a ser

processada a cada momento a um conjunto reduzido de dados. Na etapa de sumarização busca-se representar o conjunto de dados observados por meio de técnicas que forneçam uma visão global dos dados. Nessa etapa, diversos métodos podem ser empregados, tais como: 1) medidas estatísticas descritivas; 2) amostragem; 3) histogramas; 4) transformações de Fourier ou; 5) cálculo da dimensão fractal, que não é propriamente uma sumarização no sentido literal da palavra, mas sim uma medida que fornece um valor de correlação para um conjunto de dados, sendo apresentado em mais detalhes na Seção 4. A mineração dos dados, busca extrair informações dos dados delimitados e sumarizados nas etapas anteriores, por meio de métodos para extração de padrões frequentes, classificação ou agrupamento (*clustering*) dos dados (AGGARWAL, 2007; GAMA, 2010; AGGARWAL, 2013).

Os métodos de agrupamento de dados, de interesse desta tese, destinam-se a encontrar dados com comportamento ou características semelhantes e os agrupar o mais homoganeamente possível. Esses métodos podem agrupar dados de diversas maneiras, dentre elas: por meio do particionamento dos elementos (ou itens de dados), por construção de uma hierarquia de grupos, por meio de uma estrutura de grade ou baseado na densidade dos pontos. Cada uma dessas abordagens pode gerar os agrupamentos utilizando, basicamente, duas estratégias: na primeira, chamada de agrupamento de dados ou agrupamento de exemplos cada ponto de dado do fluxo de dados é associado a um grupo, de tal modo que pontos de dados pertencentes a um mesmo grupo possuam características similares, ao passo que pontos de dados pertencentes a grupos diferentes são dissimilares. Em tarefas de agrupamento de dados, cada ponto de dado, gerado por uma única fonte geradora, é analisado e agrupado.

A segunda estratégia, mais recente, é o agrupamento de fluxos de dados (CHAOVALIT, 2012; GAMA, 2010; RODRIGUES; GAMA, 2014). O agrupamento de fluxos de dados, também chamado de agrupamento de *variáveis* (CHAOVALIT, 2012), agrupamento de sensores ou, ainda, agrupamento de fluxos de dados, por objetivo identificar os sensores/fontes geradoras de dados com comportamento similar ao longo do tempo e agrupá-los. No exemplo apresentado na Figura 1 – dados gerados pelo sensor da cidade de Sete Lagoas no estado de Minas Gerais – ao ser aplicado esse tipo de técnica de agrupamento buscaria agrupar, junto ao sensor de Sete Lagoas, outros sensores que tiveram comportamento semelhante ao longo do ano de 1960. Nesta tese, será adotado o termo agrupamento de fluxos de dados, para a tarefa de agrupamento das fontes geradoras de fluxos contínuos de dados, sejam elas multidimensionais ou unidimensionais.

O agrupamento de fluxos de dados pode ser utilizado, por exemplo, na área de agrometeorologia para categorizar os regimes de chuva, para extrair índices relacionados ao clima em diferentes regiões ou, ainda, pode ser aplicado para agrupar sensores meteorológicos visando avaliar o impacto de variações climáticas extremas em infraestruturas elétricas e hidráulicas (GANGULY; STEINHAEUSER, 2008). O agrupamento de fluxos de dados também pode ser utilizado em outros domínios, por exemplo, o financeiro em que é possível agrupar ações, cotadas em uma bolsa de valores, com comportamento semelhante em um período de tempo.

Nesta tese de doutoramento, o foco do trabalho de pesquisa foi o agrupamento de fluxos de dados, em particular, fluxos de dados multidimensionais, por ser um problema ainda pouco explorado, com diversos desafios a serem tratados e potencial para contribuições relevantes ao estado da arte, como descrito na Seção 1.3.

1.3 Lacuna de Pesquisa e Objetivos do Trabalho

Muitos métodos propostos na literatura têm por finalidade realizar a extração de conhecimento de fluxos de dados utilizando a técnica de agrupamento de dados (AGGARWAL *et al.*, 2003; GUHA *et al.*, 2003; ACKERMANN *et al.*, 2012; REHMAN *et al.*, 2014; PEREIRA; MELLO, 2014; KHAN; HUANG; IVANOV, 2016; SILVA; HRUSCHKA; GAMA, 2017; HYDE; ANGELOV; MACKENZIE, 2017; ZHENG *et al.*, 2017). No entanto, ainda há poucos trabalhos voltados para agrupamento de fluxos de dados (RODRIGUES; GAMA; PEDROSO, 2008; WIDIPUTRA; PEARS; KASABOV, 2011; KRANEN *et al.*, 2011; CHAOVALIT, 2012), e apenas o método proposto por Chaovalit (2012) considera fluxos de dados multidimensionais. Muitos desses métodos realizam a construção dos agrupamentos baseando-se em medidas estatísticas descritivas dos dados, tais como: medidas de tendência central (média, mediana e moda), medidas de dispersão (valores mínimo e máximo, desvio padrão e variância) ou medidas de distribuição (achatamento e simetria da curva de distribuição).

Entretanto, diversas limitações são encontradas em vários dos métodos de agrupamento propostos no estado da arte. Essas limitações incluem:

- Tratamento a fluxo de dados unidimensional, ou seja, fluxo de dados que possua mais de um tipo de dados, é desmembrado em diversos fluxos de dados, cada um correspondente a um tipo de dado, podendo não representar de maneira adequada as inter-relações entre as dimensões.
- Mecanismo de construção do agrupamento e de cálculo de distância entre os pontos de dados são realizados utilizando medidas descritivas, que podem suavizar o comportamento dinâmico dos dados e influenciar negativamente na construção dos agrupamentos e, muitas vezes, considerando anomalias registradas na coleta dos dados;
- Aspectos metodológicos de extração de conhecimento, por exemplo, a baixa versatilidade dos métodos propostos, necessitando de muitos parâmetros de percepção dos especialistas da área para funcionar corretamente;
- Construção dos agrupamentos sem considerar a correlação que possa existir entre as dimensões do fluxo de dados, utilizando as técnicas de sumarização de modo independente em cada uma das dimensões.

Sendo assim, ao propor novos métodos, esses devem ser projetados ambicionando minimizar as limitações acima mencionadas. Para tanto, faz-se necessário a utilização de abordagens alternativas para a construção dos agrupamentos, visando substituir as medidas descritivas por valores absolutos que melhor representem as características dos dados em um instante do tempo, por exemplo, a utilização de alguma técnica de correlação entre as diversas dimensões encontradas nos fluxos de dados atuais para o período de análise.

Devido às limitações encontradas nos métodos propostos na literatura e as suposições feitas anteriormente, foram formuladas algumas questões de pesquisa que esse doutoramento buscará responder, são elas:

- 1. Considerar a correlação entre as dimensões dos fluxos de dados multidimensionais pode melhorar a coesão dos agrupamentos formados?**
- 2. Utilizar a dimensão fractal como critério de agrupamento e de semelhança entre os fluxos de dados provenientes de fontes de dados multidimensionais reais, como redes de sensores, pode gerar grupos mais coesos se comparado ao métodos que utilizam medidas descritivas tradicionais?**
- 3. É possível processar e agrupar fluxos de dados multidimensionais em tempo real utilizando a dimensão fractal?**

A partir da hipótese formulada, esta tese teve como objetivos gerais explorar e desenvolver técnicas de extração de conhecimento de fluxos de dados com foco no agrupamento de fluxos de dados, que não incorram nas limitações de domínio, dimensionalidade e complexidade de processamento. Esse conjunto de técnicas buscou, prioritariamente, contemplar características importantes de fluxos de dados, dentre as quais destacam-se a evolução ao longo do tempo, a necessidade de realizar apenas uma leitura dos dados, a capacidade finita de processamento e memória e, principalmente, lidar com a multidimensionalidade dos fluxos de dados, levando em consideração a existência de correlação entre suas dimensões, o que é inerente a dados reais. Esse último requisito, no melhor do nosso conhecimento, ainda não havia sido explorado em técnicas de agrupamento de fluxos de dados.

Assim, este trabalho de doutoramento propõe o uso da dimensão fractal para análise de correlações entre as dimensões dos pontos de dados em fluxos de dados multidimensionais. A dimensão fractal, discutida em mais detalhes no Capítulo 4, permite mensurar o comportamento não uniforme de conjuntos de dados multidimensionais, levando em consideração as correlações lineares e não-lineares existentes entre as dimensões dos dados (SCHROEDER, 1991; FALOUTSOS; KAMEL, 1994; FALOUTSOS, 2007; BARBARA; CHEN, 2009). Diversos trabalhos nas áreas de Banco de Dados e Mineração de Dados utilizam a dimensão fractal como uma medida que captura a distribuição dos dados no espaço em que estão definidos e, portanto, pode ser aplicada em processos de análise de dados (KAMEL; FALOUTSOS, 1994; LIU, 2014; SOUSA

et al., 2007b). Além disso, trabalhos na área de mineração de fluxos de dados, mostram que o conceito de dimensão fractal também pode ser utilizado para análise de comportamento de fluxos de dados multidimensionais (SOUSA *et al.*, 2007a; BARBARA; CHEN, 2009). Esses trabalhos inspiraram o uso da dimensão fractal nas soluções propostas nesta tese.

1.4 Principais Contribuições

Este trabalho de doutorado apresenta, como principais contribuições, dois novos métodos de agrupamento de fluxos de dados. O primeiro, nomeado eFCDS (*evolving Fractal Clustering of Data Streams*) utiliza a dimensão fractal para mensurar a correlação, linear ou não, existente entre as dimensões dos dados de um fluxo de dados multidimensional num período de tempo. Esta medida, calculada para cada fluxo de dados, é utilizada como critério de agrupamento de fluxos de dados com comportamentos similares ao longo do tempo. O último, nomeado eFCC (*evolving Fractal Clusters Construction*) realiza o agrupamento de fluxos de dados multidimensionais de acordo com dois critérios principais: comportamento ao longo do tempo, considerando a medida de correlação entre as dimensões dos dados de cada fluxo de dados, e a distribuição de dados em cada grupo criado, analisada por meio da dimensão fractal do mesmo.

Ambos os métodos possibilitam ainda a identificação de *outliers* e a evolução incremental dos grupos ao longo do tempo. Além disso, as soluções propostas para tratamento de correlações em fluxos de dados multidimensionais diferem dos métodos apresentados na literatura da área, que em geral utilizam técnicas de sumarização e identificação de correlações lineares aplicadas apenas em fluxos de dados unidimensionais.

Não obstante, os métodos propostos focaram na versatilidade, buscando diminuir a interferência dos especialistas do domínio. Outros aspectos, que são de fundamental importância no processo de extração de conhecimento de fluxos de dados, considerados nos métodos propostos foram: apenas considerar pontos de dados recentes (definidos pelo especialista do domínio) e realizar o agrupamento de fluxos de dados de modo incremental, buscando refletir a mudança de comportamento ao longo do tempo, com o custo computacional viável.

Por fim, este trabalho de doutorado resultou em contribuições para pesquisas realizadas em Agrometeorologia, aplicando as técnicas de mineração de fluxos de dados desenvolvidas aos dados agrometeorológicos coletados por rede de sensores e fornecidos por pesquisadores da EMBRAPA - Campinas e do CEPAGRI UNICAMP, parceiros em trabalhos de pesquisa do Grupo de Bases de Dados e Imagens (GBDI) do Instituto de Ciências Matemáticas e de Computação (ICMC) da Universidade de São Paulo (USP) descritos em: (BONES C. C.; ROMANI, 2015; BONES C. C.; ROMANI, 2016b; BONES C. C.; ROMANI, 2016a).

1.5 Organização da Tese

O restante da presente tese de doutoramento está dividida como segue:

- **Capítulo 2 Fluxos de Dados:** As limitações e desafios na mineração de fluxos de dados são apresentados mais detalhadamente nesse Capítulo. Nele serão apresentadas, também, métodos para lidar com um fluxo de dados potencialmente infinito, através da delimitação e sumarização dos mesmos.
- **Capítulo 3 Tarefas de Agrupamento em Mineração de Fluxos de Dados:** A mineração de fluxos de dados é uma área profícua e em ebulição, com diversos métodos propostos na literatura com diferentes estratégias e abordagens. Esse Capítulo abordará algumas dessas abordagens para realizar o agrupamento de fluxo de dados, assim como alguns métodos mais relevantes, no contexto dessa tese, são discutidos, apresentando seus benefícios e as limitações de cada um. Nesse Capítulo também são apresentadas alguns métodos para mensurar a qualidade dos agrupamentos obtidos por diferentes métodos de agrupamento de fluxos de dados.
- **Capítulo 4 Teoria de Fractais:** é apresentada a Teoria dos Fractais, utilizada para classificar formas geométricas com alto grau de similaridade. Essa teoria é de grande importância para o desenvolvimento dessa tese de doutoramento e foi empregada nos métodos propostos, tanto para construir os agrupamentos, quanto para analisar os grupos formados.
- **Capítulo 5 Agrupamento de fluxos de dados e detecção de outliers baseado em fractais:** são apresentados em detalhes os dois métodos desenvolvidos no decorrer desse doutorado. O primeiro, chamado eFCDS (evolving Fractal Clustering *Data streams*) buscou abordar o agrupamento de fluxos de dados realizando o cálculo de dimensão fractal de sensores, monitorando a evolução dos dados. O segundo método, chamado eFCC (evolving Fractal Clusters Construction), realizou a construção dos agrupamentos calculando a dimensão fractal do grupo em si, verificando se o conjunto de dados de um sensor poderia ser agrupado, no grupo em análise, sem modificar significativamente a dimensão fractal do grupo. O eFCC buscou lidar com fluxos de dados multidimensionais, realizando a atualização dos agrupamentos sempre que necessário.
- **Capítulo 6 Experimentos:** Os resultados alcançados pelos métodos desenvolvidos e o comparativo com os trabalhos correlatos são esmiuçados nesse Capítulo.
- **Capítulo 7 Conclusão:** Nesse Capítulo são apresentadas as conclusões alcançadas pelo desenvolvimento desta tese, as publicações geradas e também são apresentadas sugestões para trabalhos futuros.

FLUXOS DE DADOS

O crescente número de dispositivos e sensores capazes de produzir dados continuamente ao longo do tempo culmina em um volume muito grande de informações, que, por sua vez, representam um desafio para as aplicações que as utilizam. Este fluxo de dados é gerado, em teoria, infinitamente e em altas taxas de velocidade, sendo denotado, também, por *data stream*, tema deste capítulo. Neste contexto, a Seção 2.1 apresenta a definição formal de fluxos de dados e de suas propriedades. Por serem gerados infinitamente, os fluxos de dados não podem ser armazenados completamente em memória primária ou secundária, e dificilmente, podem ser lidos mais de uma vez pela aplicação que os manipula, pois a recuperação dos dados é custosa (GAMA, 2010). Desse modo, a tais aplicações são apresentadas apenas parte dos dados produzidos/recebidos em um intervalo limitado de tempo. Na Seção 2.2 é apresentada uma das técnicas que realizam essa amostragem de fluxos de dados, as janelas deslizantes. Uma das formas de se extrair informação útil desses fluxos de dados contínuos é realizar o agrupamento de fluxos de dados que possuem comportamento e/ou propriedades similares ao longo do tempo. Uma vez que o foco desta tese está no agrupamento de fluxos de dados, a Seção 2.3 apresenta as abordagens encontradas na literatura que manipulam um fluxo de dados e realizam sua sumarização. Medidas para mensurar a distância entre pontos de dados para mensurar a distância entre fluxos de dados, são apresentadas na Seção 2.3.3. Por fim, a Seção 2.4 apresenta as considerações finais do capítulo.

2.1 Conceitos de Fluxo de Dados

Um fluxo de dados é descrito como uma sequência de pontos de dados $x_1, \dots, x_i, \dots, x_\infty$ lidos em ordem crescente, a partir do índice i (GUHA *et al.*, 2003). Cada ponto x_i é gerado por uma ou mais fontes e enviado para uma ou mais entidades de processamento, tal que a ordem de chegada dos pontos de dados, em geral, não é controlada por essas entidades (GOLAB, 2009). A partir dessa descrição um fluxo de dados é formalmente expresso pela definição 1.

Definição 1 (Data stream). Um fluxo de dados $S = \{x_1, x_2, \dots, x_i, \dots\}$ é formado por uma coleção de infinitos pontos x_i , sendo que $i \in [1, \infty]$ é um instante de tempo.

A definição 1 descreve um fluxo de dados unidimensional. Esse formato de fluxos de dados é utilizado em inúmeros métodos de extração de conhecimento (CI; GUIZANI; SHARIF, 2007; BIRANT; KUT, 2007; ARANGANAYAGI; THANGAVEL, 2007; AGGARWAL *et al.*, 2007; YIN; GABER, 2008; RODRIGUES; GAMA; LOPES, 2008; RODRIGUES; GAMA; PEDROSO, 2008; GUHA, 2009; KREMER *et al.*, 2011; ERIKSSON *et al.*, 2011; WIDIPUTRA; PEARS; KASABOV, 2011; TU *et al.*, 2012; KRISHNAMURTHY *et al.*, 2012; SILVA; CHIKY; HÉBRIL, 2012; ACKERMANN *et al.*, 2012; WANG *et al.*, 2013; FORESTIERO; PIZZUTI; SPEZZANO, 2013; ZHU *et al.*, 2014; RODRIGUES; GAMA, 2014; PEREIRA; MELLO, 2014; MILLER *et al.*, 2014; LEE; LAKSHMANAN; MILIOS, 2014; CHAIRUKWATTANA *et al.*, 2014; ZHANG *et al.*, 2014; REHMAN *et al.*, 2014; KHAN; HUANG; IVANOV, 2016; PUSCHMANN; BARNAGHI; TAFAZOLLI, 2017; ZHENG *et al.*, 2017; SILVA; HRUSCHKA; GAMA, 2017; HYDE; ANGELOV; MACKENZIE, 2017). Mas apenas alguns desses métodos podem lidar com um conjunto de fluxos de dados $\mathbb{S}$, tal que $\mathbb{S} = \{S_1, S_2, \dots, S_N\}$ (RODRIGUES; GAMA; PEDROSO, 2008; WIDIPUTRA; PEARS; KASABOV, 2011). Como exemplificado pela Figura 3, várias fontes geradoras de dados (sensores) podem enviar dados para uma unidade de processamento central.

Figura 3 – Fontes geradoras de fluxos de dados enviando dados para entidade de processamento.

Fonte: Elaborada pelo autor.

Entretanto, nos dias atuais, muitos fluxos de dados geram, a cada instante de tempo i , diversos pontos de dados multidimensionais tal que a quantidade de dimensões é finita. Por exemplo, na Figura 1, um sensor gera um fluxo de dados com três dimensões: Radiação Solar,

Temperatura Máxima e Temperatura Mínima. Esse tipo de fluxo de dados, é nomeado como *fluxo de dados multidimensional* e, também conhecido como *data stream multidimensional*, é formalizado pela definição 2.

Definição 2 (Fluxo de dados multidimensional). Seja $D, 1 \leq D < \infty$, o número de dimensões; $i, 1 \leq i < \infty$, um instante de tempo, e x_i^d o valor de um ponto de dado x no tempo i para a dimensão $d, 1 \leq d \leq D$. Um fluxo de dados multidimensional é dado por $DS = \{x_1^1, x_1^2, \dots, x_1^D, \dots, x_2^1, \dots, x_2^D\}$, ou seja, $DS = [x_i^d]$.

Dadas as duas definições anteriores, é frequentemente encontrada na literatura algumas características importantes inerentes a fluxos de dados e que devem ser consideradas, conforme apresentado em (BABCOCK *et al.*, 2002; GUHA *et al.*, 2003; GOLAB, 2009; GAMA, 2010; CHAOVALIT, 2012; AGGARWAL, 2013), a saber:

- O fluxo de dados é contínuo e potencialmente infinito;
- Os dados podem ser multidimensionais;
- Os dados podem evoluir com o passar do tempo;
- Uma vez realizada a leitura dos dados, eles são descartados ou arquivados, sendo não-trivial realizar uma segunda leitura.

Com características tão peculiares, o processamento e a extração de conhecimento de fluxos de dados devem ser tratados de maneira diferenciada se comparados ao processamento e à extração de conhecimento em bancos de dados convencionais. Por exemplo, dentre as diferenças importantes no gerenciamento dos dados e na busca por informações em um *Database Management System* (DBMS) e em um *Data Stream Management Systems* (DSMS), pode-se destacar (BABCOCK *et al.*, 2002; GAMA, 2010): nos DBMS as consultas são executadas de uma só vez e o acesso é randômico e quase direto aos dados, sendo regidos por estruturas de indexação (ELMASRI; NAVATHE, 2011); por outro lado, nos DSMS, os fluxos de dados são transientes, as consultas são contínuas e o acesso aos dados é, em geral, realizado de maneira sequencial (GAMA, 2010).

Dadas essas considerações, algumas perguntas surgem:

- Como realizar apenas uma leitura dos dados em fluxos de dados?
- Como é possível manipular um grande volume de dados que chega continuamente?
- Se os dados chegam a uma velocidade maior que a capacidade do servidor de processá-los, como é possível fazer tal processamento?
- Como acompanhar a característica evolutiva dos fluxos de dados?

- Como extrair padrões de interesse e conhecimento relevante de fluxos de dados, considerando suas características?

Buscando responder algumas dessas perguntas, vêm sendo desenvolvidos métodos e técnicas capazes de sumarizar, restringir o tamanho da leitura e extrair conhecimento de fluxos de dados. Alguns desses métodos e técnicas, de interesse no contexto deste projeto, são apresentados nas seções seguintes. Na Seção 2.3 são apresentadas as técnicas de sumarização de fluxos de dados. Alguns métodos de delimitação dos fluxos de dados estão apresentados na Seção 2.2. Algumas medidas para mensurar a distância entre pontos de dados e o método *Dynamic Time Warping*, utilizado para mensurar a distância entre fluxos de dados, são apresentados na Seção 2.3.3.

2.2 Delimitação de Fluxo de Dados

Em muitas aplicações, a meta é tomar decisões baseando-se em estatísticas ou modelos obtidos sobre dados que estão sempre disponíveis. Entretanto, no contexto de fluxos de dados os dados chegam para serem processados continuamente, tornando impraticável, em muitos casos, o armazenamento de todo o fluxo de dados, bem como realizar mais de uma leitura do mesmo. Desse modo, uma estratégia bastante usual é delimitar o tamanho do conjunto de dados observado utilizando janelas deslizantes (*sliding windows*) (DATAR *et al.*, 2002; GABER; ZASLAVSKY; KRISHNASWAMY, 2005; AGGARWAL; YU, 2007; GAMA, 2010; BRAVERMAN; OSTROVSKY; ZANIOLO, 2012; LI; LEE, 2009).

Janelas deslizantes é uma técnica utilizada para restringir o tamanho do conjunto de dados a ser analisado, observando-se apenas os N pontos de dados mais recentes (BIFET; GAVALDA, 2007). Uma janela deslizante pode ser considerada uma janela ativa de pontos de dados, em um determinado instante de tempo (DATAR *et al.*, 2002).

Janelas deslizantes podem ser utilizadas, por exemplo, para manter estatísticas sobre pacotes de redes processados por um conjunto de roteadores nas últimas 24 horas. Estas estatísticas são obtidas de maneira contínua em intervalos regulares de tempo, de acordo com o tamanho da janela. Em Nunes *et al.* (2013) os autores utilizaram janelas deslizantes para continuamente computar estatísticas de fluxos de dados provenientes de sensores agrometeorológicos.

Para realizar essa delimitação as janelas deslizantes contam com duas abordagens básicas de tamanho fixo e outra estratégia em que é fixado um marco inicial no tempo para início da janela (BIFET; GAVALDA, 2007). As duas abordagens para janelas de tamanho fixo são: a primeira, observa uma determinada quantidade N de dados pré-estabelecida, e a segunda, baseada em tempo observa os pontos de dados por um período determinado de tempo (BIFET; GAVALDA, 2007; DATAR *et al.*, 2002; ZHOU *et al.*, 2008).

O modelo de janela deslizante de tamanho fixo, considerado o tipo de janela deslizante

mais simples, é baseado em uma sequência de dados em que a janela é definida em relação ao número de observações que se deseja (GAMA, 2010). Pode-se, por exemplo, manter observações de um tamanho j qualquer. Esse tipo de janela é semelhante à estrutura de dados FIFO - *First In First Out*, pois, quando um ponto de dado j entra na janela de tamanho ω , um ponto de dado $j - \omega$ é eliminado da janela.

No modelo de janelas deslizantes baseadas em tempo, o tamanho da janela é definido em relação à duração T da janela (GOLAB, 2006) e todos os pontos de dados que estiverem dentro do período de tempo T da janela são observados (GAMA, 2010). A Figura 4 exemplifica uma janela deslizante em que a quantidade observada de dados possui tamanho fixo.

Figura 4 – Exemplo de janela deslizante de quantidade fixa de dados

Fonte: Elaborada pelo autor.

O outro tipo de janela deslizante, em que é fixado um marco inicial para início da janela, é a *Landmark Windows*. Esse tipo de janela deslizante é utilizado, por exemplo, quando se deseja observar algum evento específico que ocorre nos dados, por um determinado período de tempo. Um exemplo de *Landmark Windows* é apresentado na Figura 5.

Um modelo de janela deslizante mais recente é o que mantém uma janela de tamanho variável, como no modelo ADWIN (*Adaptive sliding window*) proposto por Bifet e Gavalda (BIFET; GAVALDA, 2007). Nesse trabalho os autores utilizam as janelas deslizantes cujo tamanho, em vez de ser fixado *a priori*, é recalculado em tempo real de acordo com a taxa de variação observada a partir dos dados da própria janela. No ADWIN as janelas crescem automaticamente quando os dados são estacionários, para maior precisão, e encolhem automaticamente quando uma mudança está ocorrendo, para descartar dados desatualizados (BIFET; GAVALDA, 2007).

Figura 5 – Exemplo de janela deslizante com início fixado no tempo - *Landmark Windows*

Fonte: Elaborada pelo autor.

2.3 Sumarização de fluxos de dados

A utilização de sinopses ou sumarizações é essencial para a gestão dos dados massivos em grandes conjuntos de dados como fluxos de dados (CORMODE; HADJIELEFTHERIOU, 2010). Nos trabalhos de Cormode e Hadjieleftheriou (2010) e Gama (2010) são apresentadas diversas técnicas e métodos para sumarizar fluxos de dados, tais como:

- As técnicas de histograma, que buscam representar a frequência de distribuição dos valores dos pontos de dados, fornecendo uma sumarização do comportamento dos dados. Muitas técnicas de histograma foram propostas na literatura, dentre as quais estão: *Equi-Width Histograms*, *Equi-Depth Histograms*, *Singleton-Bucket Histograms*, *Lattice Histograms* (CORMODE *et al.*, 2012; CORMODE; HADJIELEFTHERIOU, 2010);
- Técnicas de amostragem, que em oposição aos métodos que trabalham com todos os dados, elas selecionam um pequeno subconjunto de dados para serem analisados em intervalo regular. Dentre as técnicas de amostragens têm-se: *Simple Random Sampling With Replacement*, *Poisson Sampling* e *Stratified Sampling*;
- *Sketches* são utilizadas para sumarizar os dados aplicando-se, geralmente uma função de transformação nos dados de entrada. Diversos métodos realizam *Sketch* em fluxos de dados, por exemplo, *Count-Min Sketch*, *K-ary Sketch* e *AMS Sketch*. Diferentemente das técnicas de amostragem, que visualizam apenas uma porção dos dados de entrada, as técnicas de *Sketch* visualizam todos os dados, retraindo apenas uma pequena sumarização deles (CORMODE *et al.*, 2012).
- *Wavelets* são como transformadas de Fourier que buscam capturar a evolução ou as tendências de funções numéricas realizando a decomposição do sinal em um conjunto de

coeficientes (GILBERT *et al.*, 2003), em que é possível citar as *Haar Wavelets* e *Restricted Haar Wavelets*.

Algumas das técnicas comuns em trabalhos de sumarização para dados tradicionais e fluxos de dados são histogramas e *Wavelets*, apresentados a seguir.

2.3.1 Histograma

Histograma é uma das técnicas que pode ser utilizada na sumarização de fluxo de dados. Um histograma busca representar a frequência de distribuição dos valores dos pontos de dados (GAMA, 2010) ou de um conjunto de ponto de dados do fluxo de dados (CORMODE *et al.*, 2012).

Diversos tipos de histogramas podem ser encontrados, cada qual representa uma determinada característica sobre o comportamento dos dados (CORMODE *et al.*, 2012). Dois tipos básicos de histogramas são descritos a seguir:

- Histograma simétrico: nesse tipo de histograma a frequência é mais alta no centro e decresce gradualmente para as caudas de maneira simétrica, conforme apresenta a Figura 6(a). A média e a mediana são aproximadamente iguais e localizam-se no centro do histograma, onde está localizado o pico. Esse tipo de histograma ocorre, geralmente, quando os dados são obtidos de processos padronizados e estáveis (JAWERTH; SWELDENS, 1994; CORMODE *et al.*, 2012).
- Histograma assimétrico: a frequência decresce bruscamente em um dos lados e de forma gradual no outro, produzindo uma calda mais longa em um dos lados. A média localiza-se fora do meio da faixa de variação. Quando a assimetria é à direita a mediana é inferior à média, conforme apresenta a Figura 6(b). Quando a assimetria é à esquerda a mediana é superior à média (JAWERTH; SWELDENS, 1994; CORMODE *et al.*, 2012).

Guha, Shim e Woo (2004) desenvolveram o método *V-Optimal Histogram* que constrói um histograma unidimensional que minimiza as medidas relativas dos erros. Outro método proposto para prover histogramas de fluxo de dados é o *cluster histograms* (AGGARWAL, 2012), desenvolvido para estimar e resumir os perfis de distribuição de dados mais importantes sobre diferentes segmentos do fluxo de dados.

Diversos estudos e aplicações de histogramas podem ser encontrados em (CORMODE *et al.*, 2012; GUHA; SHIM; WOO, 2004; GAMA, 2010; GUHA; KOUDAS; SHIM, 2006; GUHA, 2009), dentre os quais encontram-se histogramas bidimensionais, multidimensionais, *Equi-Width Histograms*, *Equi-Depth Histograms*, *Singleton-Bucket Histograms* e *Lattice Histograms*, sendo que todos esses tipos de histogramas podem ser adaptados para fluxos de dados. Abaixo é apresentado como um *Equi-Width Histograms* pode ser facilmente construído:

Figura 6 – Exemplo de histogramas: (a) simétrico, com o pico localizado no centro do gráfico; (b) assimétrico, com o pico deslocado do centro

Fonte: Elaborada pelo autor.

- *Equi-Width Histograms* - considerado um dos mais fáceis de construir (CORMODE *et al.*, 2012), cria *buckets*, que são intervalos de valores aos quais são associados cada um dos dados de entrada. Dado o vetor $V = 1, 2; 1, 4; 2, 1; 3, 4; 3, 5; 3, 4; 2, 3; 4, 3; 4, 4; 4, 2; 4, 4; 4, 5; 1, 5$, cada ponto de dados é atribuído a um de quatro *buckets*, ou subconjuntos disjuntos, denotados por $S1$, $S2$, $S3$, e $S4$, conforme pode ser observado na Figura 7. Para o vetor V os *buckets* correspondem aos quatro subintervalos disjuntos do domínio de valor de dados $[1, 5]$, que são apresentados no eixo horizontal. Então, cada ponto de dados é atribuído ao *bucket* cujo subintervalo contém o ponto. A altura de cada barra de histograma corresponde ao número de pontos que caem no *bucket* correspondente, isto é, a frequência de pontos associados ao *bucket*. Esse histograma é determinado no limite inferior pelo *bucket* (1,0) da esquerda, a largura do *bucket* (1,0), e as quatro contagens de frequência para os quatro recipientes (3, 2, 3, 5). Embora, nesse exemplo, os dados foram reduzido de treze números para seis números (zero a cinco), é claro que o grau de compressão pode ser muito maior, por exemplo, um histograma semelhante representando 200 pontos com valores entre um e cinco ainda seria representado por seis números.

O uso de histogramas em fluxos de dados pode ser realizado em conjunto com as janelas deslizantes apresentadas na seção 2.2, dessa maneira, o cálculo dos *buckets* e a atribuição dos valores a cada *bucket* ocorre utilizando-se o limite de dados imposto pela janela deslizante.

2.3.2 Wavelets

Similares às transformadas de Fourier, *wavelets* são transformadas que buscam capturar a evolução ou as tendências de funções numéricas realizando a decomposição do sinal em um conjunto de coeficientes (GILBERT *et al.*, 2003). O sinal pode ser reconstruído utilizando-se todos os coeficientes do conjunto sem perda significativa de informação (GAMA, 2010).

Figura 7 – Histograma width, com *buckets* construído com base nos limites inferior e superior do vetor $V = 1, 2; 1, 4; 2, 1; 3, 4; 3, 5; 3, 4; 2, 3; 4, 3; 4, 4; 4, 2; 4, 4; 4, 5; 1, 5$.

Fonte: Elaborada pelo autor.

As *wavelets* são conceitualmente muito similares aos histogramas. A principal diferença é que, enquanto histogramas produzem *buckets* que são subconjuntos do domínio de dados do atributo original, as *wavelets* são transformações dos dados que tentam representar a característica mais significativa do domínio (isto é, a frequência), podendo capturar combinações de informações de alta e baixa frequência (CORMODE *et al.*, 2012).

Uma grande vantagem das *wavelets*, que a torna muito interessante para algoritmos voltados para fluxos de dados, é a possibilidade de realizar a reconstrução do sinal, quase que na sua íntegra, mesmo com a ausência ou eliminação de coeficientes não significativos.

Existem diversos tipos de transformadas de *wavelets*, como as ortogonais, polinomiais e multidimensionais (JAWERTH; SWELDENS, 1994; CORMODE *et al.*, 2012). Um dos tipos de *wavelets* mais empregados para a decomposição do sinal de fluxos de dados é a *Haar wavelet* (HWT - Haar Wavelet Transform), devido ao seu conjunto reduzido de coeficientes produzidos (AGGARWAL; YU, 2007; GAMA, 2010; CORMODE *et al.*, 2012).

Dado um vetor unidimensional $A = [2, 2, 0, 2, 3, 5, 4, 4]$, contendo $N = 8$ valores, a HWT pode ser computada da seguinte maneira. Primeiro obtém-se a média dos valores par a par, $[(2 + 2)/2, (0 + 2)/2, (3 + 5)/2, (4 + 4)/2]$, gerando uma nova representação de "baixa resolução" dos dados, com os seguinte valores médios $[2, 1, 4, 4]$. Obviamente, alguma informação foi perdida no vetor que contém as médias. Para ser capaz de restaurar os valores originais dos dados (tratados como uma matriz de dados ou vetor), é preciso armazenar alguns coeficientes de detalhe, para capturar as informações que faltam (GAMA, 2010; JAWERTH; SWELDENS, 1994). Na HWT, esses coeficientes de detalhe são as diferenças das médias calculadas para o segundo valor do par. Assim, no exemplo apresentado anteriormente, a primeira média calculada é 2 e o segundo valor do primeiro par de ponto de dados do vetor original também é 2, então o coeficiente de detalhe é 0, pois $2 - 2 = 0$. Ao calcular o segundo coeficiente obtém-se o valor -1 , haja vista que a segunda média calculada é 1 e o segundo valor do segundo par de ponto de dados do vetor original é 2, logo $1 - 2 = -1$. A Figura 8 mostra o resultado da transformação de Haar

aplicada ao vetor A . É importante salientar que nenhuma informação foi perdida neste processo, bastando utilizar o vetor que contém as médias e o vetor dos coeficientes para reconstruir o sinal original. Desse modo, a HWT do vetor A , W_A , é dada pela média geral de todos os pontos de dados, seguido de todos os coeficientes obtidos na transformada do mais genérico até os mais detalhados, $W_A = [11/4, -5/4, 1/2, 0, 0, -1, -1, 0]$

Figura 8 – Exemplo da aplicação da transformada de Haar ao vetor A

	Vetor Original e médias	Vetor de Coeficientes
3	[2, 2, 0, 2, 3, 5, 4, 4]	---
2	[2, 1, 4, 4]	[0, -1, -1, 0]
1	[3/2, 4]	[1/2, 0]
0	[11/4]	[-5/4]

Fonte: Elaborada pelo autor.

É possível notar, de forma intuitiva, que os coeficientes da *wavelet* possuem pesos diferentes no que diz respeito à sua importância na reconstrução dos valores de dados originais. Por exemplo, a média global é, obviamente, mais importante do que qualquer coeficiente de detalhe, uma vez que afeta a reconstrução de todas as entradas de dados. De forma a equalizar a importância de todos os coeficientes da *wavelet*, é necessário normalizar as entradas finais da W_A adequadamente (JAWERTH; SWELDENS, 1994; CORMODE *et al.*, 2012). Uma normalização comum é aplicar a cada coeficiente da *wavelet* o produto por $\sqrt{M/2} = 2^{(\log M - l)/2}$, onde M é o número de valores de entrada e l representa o nível em que o coeficiente aparece ($l = 0$ indica o menor nível). Desse modo, a normalização do coeficiente i da HWT $c_i = c_i \sqrt{\frac{M}{2^{\text{nível}(c_i)}}}$ garante a ortonormalidade da *wavelet* básica de Haar (JAWERTH; SWELDENS, 1994).

2.3.3 Funções de Distância

Para realizar o agrupamento vários métodos utilizam medidas de distância entre os pontos de dados do fluxo de dados ou entre sequências de pontos de dados delimitadas por janelas. Dentre as funções de distância encontradas na literatura frequentemente utilizadas para mensurar a distância entre fluxos de dados, destacam-se (TOYODA; SAKURAI; ISHIKAWA, 2013): a Distância Euclidiana (DE) (ZHANG; LIVNY, 1996; CAO *et al.*, 2006; BERINGER; HÜLLER-MEIER, 2006), a *Dynamic Time Warping* (DTW) (TOYODA; SAKURAI; ISHIKAWA, 2013; NIRJON *et al.*, 2014) e a distância baseada na Maior Subsequência Comum (*Longest Common SubSequence*- LCS) (LIBEN-NOWELL; VEE; ZHU, 2006; WANG; KORKIN; SHANG, 2011). Cada uma dessas funções de distância, utilizadas para mensurar quão similar ou dissimilar são

dois fluxos de dados, são apresentadas, brevemente, a seguir, despendendo maior foco para a DTW por comparar fluxos de dados como um todo e não ponto a ponto como a DE ou em subsequências como a LCS.

- **Distância Euclidiana:** a Distância Euclidiana (DE) busca mensurar o quão afastado estão dois pontos no plano cartesiano (ZHANG; LIVNY, 1996). Todavia, ela também pode ser utilizada para realizar o cálculo da distância entre os pontos de dados de dois fluxos de dados $A = (a_1, a_2, \dots, a_\infty)$ e $B = (b_1, b_2, \dots, b_\infty)$ (CAO *et al.*, 2006; BERINGER; HÜLLERMEIER, 2006). Para calcular a DE é necessário utilizar a fórmula:

$$DE(A, B) = \sqrt{\sum_{i=1}^W (a_i - b_i)^2} \quad (2.1)$$

em que, a_i e b_i são pontos de dados pertencentes aos fluxos de dados A e B , respectivamente e, W representa o tamanho da janela deslizante, ou seja, a quantidade de pontos a_i e b_i que podem ser observados na janela deslizante.

- **Longest Common SubSequence:** inicialmente desenvolvida para verificar se uma subsequência poderia ser encontrada em um texto (WU; SALZBERG; ZHANG, 2004), a *Longest Common SubSequence* (LCS) foi adaptada para encontrar um subsequência que seja comum entre dois fluxos de dados e, se existir, qual é a maior subsequência possível (LIBEN-NOWELL; VEE; ZHU, 2006; WU *et al.*, 2005). Diferentemente do problema de encontrar uma subsequência pré-definida em um texto, encontrar a LCS entre dois fluxos de dados, sem fornecer, à priori, a subsequência é um trabalho considerado NP-complexo (LIBEN-NOWELL; VEE; ZHU, 2006). Existem muitas pesquisas na literatura para encontrar a LCS em fluxos de dados, dentre as quais estão (LIBEN-NOWELL; VEE; ZHU, 2006; WANG; KORKIN; SHANG, 2011; WU; SALZBERG; ZHANG, 2004; WU *et al.*, 2005; DING *et al.*, 2008).
- **Dynamic Time Warping:** *Dynamic Time Warping* (DTW), é uma transformação que permite esticar sequências ao longo do eixo do tempo a fim de minimizar a distância entre elas (TOYODA; SAKURAI; ISHIKAWA, 2013). Para realizar o cálculo da DTW entre dois fluxos de dados $A = (a_1, a_2, \dots, a_\infty)$ e $B = (b_1, b_2, \dots, b_\infty)$, constrói-se uma matriz $N \times M$, onde os ponto de dados (i, j) da matriz correspondem ao quadrado da distância entre os pontos de (A, B) , sendo, $d(a_i, b_j) = (a_i - b_j)^2$, o alinhamento entre os pontos a_i e b_j . Para encontrar a melhor correspondência entre essas duas sequências, deve-se encontrar um caminho através da matriz que minimiza a distância total acumulada entre eles. Um caminho de deformação, W , é um conjunto contíguo de ponto de dados da matriz que caracteriza um mapeamento entre A e B , conforme apresentado na Figura 9. O k elemento de W é definido como $w_k = (i, j)_k$. Portanto,

$$W = w_1, w_2, \dots, w_k, \dots, w_K \max(n, m) \leq K \leq n + m - 1 \quad (2.2)$$

E, por definição, o caminho ótimo W_0 é o caminho que possui a menor soma dentre todos os caminhos, dado pela fórmula:

$$DTW(A, B) = \min \left(\sqrt{\sum_{k=1}^K W_k} \right) \quad (2.3)$$

Figura 9 – Exemplo construção e identificação do caminho mínimo na Matriz gerado pela técnica DTW.

Fonte: Elaborada pelo autor.

A Figura 10 apresenta os dois fluxos de dados (A, B) alinhados pela DTW.

Figura 10 – Representação gráfica do alinhamento obtido pela DTW de dois fluxos de dados (A, B) .

Fonte: Elaborada pelo autor.

A DTW foi, também, discutida e aperfeiçoada em (RAKTHANMANON *et al.*, 2012) para séries temporais de trilhões de dados, tendo enfoque em um método de busca exata através da DTW. Nesse trabalho os pesquisadores propuseram quatro otimizações baseadas na parada antecipada do processamento. As melhorias fundamentam-se nas técnicas de *Early Abandoning* e em *Lower Bounds*. *Early Abandoning* é a parada antecipada do processamento ao se constatar que os valores obtidos estão se distanciando do valor

esperado (KEOGH *et al.*, 2006). No caso da DTW o processamento é interrompido quando a soma dos quadrados das distâncias está se afastando do valor esperado. Já *Lower Bounding* é uma técnica que se utiliza para eliminar candidatos que estejam abaixo de um limite inferior (SHOUP, 1997). No trabalho apresentado em (RAKTHANMANON *et al.*, 2012), o método de *Lower Bound* é utilizado para eliminar os pontos que não atinjam um valor mínimo para serem processados.

Segundo (RATANAMAHATANA; KEOGH, 2004) a DTW produz melhores resultados na construção de agrupamento de séries temporais do que utilizar a DE. Tendo em vista que séries temporais são considerados fluxos de dados de tamanho fixo (GAMA, 2010), então seu uso em fluxos de dados, também poderá produzir resultados melhores que a DE, pois as sequências são ponto de dados do fluxo de dados delimitados por janelas deslizantes. Entretanto, o custo quadrático da DTW, em relação a quantidade de dados de entrada, representa um empecilho quando utilizada em fluxos de dados que possuem frequência de geração dos dados elevada. Por esse motivo, o uso de dimensão fractal torna-se uma opção mais viável, por possuir complexidade computacional menor e acompanhar a evolução presente nos dados.

2.4 Considerações finais

Diversas técnicas têm sido desenvolvidas para restringir o tamanho de fluxos de dados, permitindo que se trabalhe com uma porção dos dados por vez. É possível encontrar, também, diversos métodos capazes de sumarizar fluxos de dados, contribuindo para o uso limitado dos recursos computacionais disponíveis, bem como, possibilitando que fluxos de dados potencialmente infinitos sejam processados ao se utilizar o conceito de janelas deslizantes.

Algumas das técnicas mencionadas nesse capítulo foram utilizadas para auxiliar a construção dos métodos de mineração de fluxos de dados propostos nesta tese, buscando restringir e sumarizar a quantidade de dados que são processados.

No Capítulo 3 são apresentados métodos capazes de realizar a mineração de fluxo de dados, com foco nos métodos que realizam o agrupamento de um conjunto de fluxos de dados (S).

TAREFAS DE AGRUPAMENTO EM MINERAÇÃO DE FLUXOS DE DADOS

A extração de conhecimento a partir de grandes volumes de dados foi definido em (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996) como um processo não trivial de identificação de padrões embutidos nos dados que sejam válidos, novos, potencialmente úteis e essencialmente compreensíveis. O processo de extrair conhecimento de grandes bases de dados inclui as etapas de seleção de dados, processamento, transformação, mineração de dados e interpretação (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996; GOLAB, 2009). Dentre essas etapas, a de mineração de dados busca explorar e analisar grandes quantidades de dados para descobrir padrões e regras significativas (GOLAB, 2009).

Minerar dados para extração de conhecimento em conjuntos de dados convencionais por si só é uma tarefa dispendiosa (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996). No contexto de fluxos de dados, métodos para minerar e extrair conhecimento devem levar em consideração alguns requisitos adicionais, dentre os quais pode-se destacar:

- A necessidade de ler os dados apenas uma vez (BABCOCK *et al.*, 2002; AGGARWAL *et al.*, 2007; GAMA, 2010): devido ao grande volume de dados e à capacidade finita de processamento e memória, torna-se necessário que todo o método desenvolvido para minerar dados em fluxo de dados realize apenas uma leitura dos dados. Haja visto que múltiplas leituras de dados que chegam continuamente, em grande velocidade e volume, é impraticável.
- Extrair e manter um conjunto de padrões atualizado (BARBARA, 2002; GAMA, 2012): para acompanhar a característica evolutiva dos dados é primordial que o modelo de mineração mantenha um conjunto de padrões atualizado, buscando representar de modo mais fiel possível os dados já processados.

- Capacidade de encontrar padrões desconhecidos (GAMA, 2012; CHAOVALIT, 2009): outra característica que os métodos de mineração em fluxo de dados devem possuir é a capacidade de encontrar padrões que não correspondem aos padrões atuais de representação, mas que possam vir a ser frequentes no futuro.

Diversos métodos têm sido desenvolvidos para minerar dados em fluxos de dados, cada qual com estratégias e focos diferentes, tais como:

- Mineração de padrões frequentes (*Frequent pattern mining*) em (CHARIKAR; CHEN; FARACH-COLTON, 2002; CORMODE; HADJIELEFTHERIOU, 2010; LI; LEE, 2009; TOYODA; SAKURAI; ISHIKAWA, 2013);
- Classificação em (AGGARWAL *et al.*, 2004; GABER; ZASLAVSKY; KRISHNASWAMY, 2007; GABER; ZASLAVSKY; KRISHNASWAMY, 2005);
- Detecção de *outliers* em (BARBARA, 2002);
- Agrupamentos em (AGGARWAL *et al.*, 2003; BERINGER; HÜLLERMEIER, 2006; GAMA *et al.*, 2009; HAHSLER; DUNHAM, 2011; FELDMAN; LANGBERG, 2011; KRISHNAMURTHY *et al.*, 2012; ACKERMANN *et al.*, 2012; WANG *et al.*, 2013; FORRESTIERO; PIZZUTI; SPEZZANO, 2013; ZHU *et al.*, 2014; CHAIRUKWATTANA *et al.*, 2014; DING *et al.*, 2016; HYDE; ANGELOV; MACKENZIE, 2017; SILVA; HRUSCHKA; GAMA, 2017).

Nessa tese de doutoramento, o foco foi na tarefa de agrupamento de fluxos de dados. Por esse motivo, as seções seguintes abordam as diferentes estratégias de agrupamento em geral e específicas de agrupamento de fluxos de dados, apresentando alguns métodos, bem como seus pontos positivos e negativos.

3.1 Técnicas de Agrupamento Aplicadas a Fluxos de Dados

Agrupamento (*Clustering*) é uma técnica utilizada amplamente em mineração de dados para agrupar pontos de dados, de modo que objetos dentro de um mesmo grupo (*cluster*) devem possuir características e propriedades muito semelhantes (AGGARWAL *et al.*, 2006), características essas que diferem de grupo para grupo. Entretanto, realizar o agrupamento de características similares pode ser muito custoso (ERIKSSON *et al.*, 2011) e até mesmo levar a resultados pouco consistentes (FAYYAD; PIATETSKY-SHAPIO; SMYTH, 1996). A Figura 11, que apresenta o particionamento de três tipos de dados diferentes (estrela, círculo e pentágono), sendo que o grupo *B* está inconsistente de acordo com as formas geométricas. Segundo Gama

(2010) o agrupamento é utilizado para se obter uma visão sobre a distribuição dos dados e como pré-processamento para outros algoritmos.

Figura 11 – Exemplo de agrupamento de ponto de dados com resultado inconsistente no grupo B.

Fonte: Elaborada pelo autor.

Alguns requisitos fundamentais, apresentados em (BARBARA, 2002), devem estar presentes nos algoritmos de agrupamentos de pontos de dados, além dos apresentados na introdução desse capítulo. São eles:

- Uma representação de tamanho compacto (GUHA *et al.*, 2003; GAMA, 2010): ao realizar o agrupamento, o espaço tomado para representar os grupos deverá, preferencialmente, não exceder a memória principal, para evitar o acesso excessivo a disco.
- Processamento rápido e incremental dos novos pontos de dados (CHAOVALIT, 2009; GAMA, 2010): o tempo gasto na atualização dos resultados e do modelo de agrupamento não deve exceder a taxa de chegada dos dados. Deste modo, os métodos de agrupamento de ponto de dados precisam despende, no máximo, uma pequena porção relativa da frequência de chegada dos dados com o seu processamento.
- Rastreabilidade das mudanças ocorridas nos grupos (GAMA, 2010): deve ser possível identificar quais os grupos foram criados, quais deixaram de existir e quais persistem ao longo do tempo.
- Identificar e reagir a mudanças (CHAOVALIT, 2009; GAMA, 2010): muitas das aplicações do mundo real geram dados de maneira não estacionária. Deste modo, é necessário identificar quando as mudanças ocorrem, buscando adequar os grupos existentes à nova realidade. Métodos tradicionais de mineração implementam a detecção de mudanças visualizando todos os dados de uma única vez (CHAOVALIT, 2009). Entretanto, os métodos de mineração em fluxos de dados devem identificar as mudanças ocorridas, visualizando apenas a porção mais recente dos dados.

- Rápida e clara identificação de exceções (*outliers*) (GAMA, 2010; GAMA; SEBASTIÃO; RODRIGUES, 2013): identificar *outliers* para que esses não influenciem demasiadamente os grupos identificados.
- Capacidade de lidar com dados multidimensionais (CHAOVALIT, 2012).

De acordo com os requisitos apresentados acima, qualquer novo método de agrupamento de fluxo de dados deve ser adaptado para realizar o agrupamento de maneira contínua, evolutiva, concisa e *online* da sequência de entrada observada, considerando suas várias dimensões quando existir.

Além disso, a característica temporal do fluxo de dados também deve ser considerada (HAHSLER; DUNHAM, 2011), utilizando uma pequena quantidade de espaço de armazenamento (GUHA *et al.*, 2003) e de tempo de processamento. Tais requisitos são impostos para comportar a maneira contínua em que os dados chegam e a necessidade de analisá-los em tempo real (GAMA, 2010).

Para cumprir os requisitos mencionados no início dessa seção, foram desenvolvidas diversas estratégias de construção dos agrupamentos, nas quais, em geral, os algoritmos de agrupamento se enquadram. Algumas dessas estratégias de construção dos agrupamentos são: por particionamento dos dados, criando uma hierarquia de grupos, baseadas em uma estrutura de grade, agrupando os fluxos de dados, dentre outras. Outra característica dos métodos de agrupamento de ponto de dados é o de utilizar medidas que quantificam a distância entre pontos de dados, por exemplo a distância de Manhattan (L_1) ou a distância Euclidiana (L_2), para então agrupar os pontos que apresentarem a menor distância (os mais similares). No entanto, seu uso com fluxos de dados requer algumas modificações, pois seu algoritmo básico requer espaço de armazenamento quadrático em relação à entrada de dados (GAMA, 2010).

O restante desse Capítulo está dividido da seguinte forma: a Seção 3.1.1 apresenta a estratégia de agrupamento hierárquico de dados e; na Seção 3.1.2 é apresentada a estratégia de agrupamento de dados baseada em partição e alguns métodos que se enquadram nesse conceito; a noção de agrupamento baseado em uma estrutura de grade é apresentada na Seção 3.1.3; na Seção 3.1.4 é apresentado o conceito de agrupamento de fluxos de dados juntamente com alguns métodos correlatos aos métodos que serão apresentados nesse trabalho. Na Seção 3.2 são apresentadas técnicas para mensurar a qualidade dos resultados obtidos pelos métodos de agrupamento em geral, quando não há rótulos nos dados e quando existe tal informação.

3.1.1 Agrupamentos Hierárquicos

Agrupamento hierárquico (*Hierarchical Clustering*) é uma abordagem de análise e mineração de dados que busca construir uma hierarquia entre os grupos. Essa hierarquia entre os grupos pode ser obtida por intermédio de um particionamento recursivo dos dados em 2, 3, 4, até

n grupos, para n pontos de dados de entrada (DASGUPTA; LONG, 2005). Nessa abordagem, os grupos intermediários surgem com o refinamento da hierarquia de grupos iniciais até se obter o *cluster* raiz, conforme apresentado no exemplo da Figura 12, onde os pontos de dados dos fluxos de dados mais similares se agrupam, por exemplo, B é mais similar a C, E com F e G sendo mais similar a H e posteriormente são formados os grupos ABC, EFGH e assim sucessivamente, até formar toda a hierarquia. Em (KRISHNAMURTHY *et al.*, 2012) os autores apresentaram o ACTIVECLUSTER, um algoritmo que utiliza essa técnica recursiva para criar agrupamentos hierárquicos com número reduzido de cálculos de similaridade entre os *clusters*, a fim de se obter um melhor desempenho.

Figura 12 – Hierarquia de grupos, onde os pontos de dados provenientes do fluxo de dados B são mais similares aos pontos de dados do fluxo de dados C, do mesmo modo E é mais similar a F e G possui similaridade acentuada com H, com o refinamento da hierarquia ABC e EFGH são agrupados, posteriormente D é agrupado a ABC e finalmente a hierarquia é completada gerando o grupo ABDCEFGH.

Fonte: Elaborada pelo autor.

Os algoritmos hierárquicos possuem três grandes vantagens sobre métodos de particionamento fluxo de dados (RODRIGUES; GAMA; PEDROSO, 2006):

- não requerem do usuário um número pré-definido de grupos a serem obtidos;
- não fazem nenhuma suposição sobre a distribuição dos dados;
- não necessitam de nenhuma representação explícita, pois ao invés de criar uma matriz de diferenças entre todos pares de ponto de dados, eles realizam o cálculo das diferenças apenas sobre pontos de dados do mesmo grupo e assim definem a hierarquia de *clusters*.

Um dos primeiros algoritmos a construir agrupamento hierárquico foi o BIRCH (ZHANG; LIVNY, 1996), que constrói uma estrutura hierárquica chamada *Clustering Feature Tree*, em que cada nó da árvore consiste em uma tripla de *Clustering Features* (CF). Um CF é caracterizado como um vetor $(N, \vec{LS}, SS)$, em que N é o número de dados no grupo, $\vec{LS}$ é a soma dos N dados, ou seja, $\sum_{i=1}^N \vec{X}_i$, e SS é a soma dos quadrados dos N dados do grupo, isto é, $\sum_{i=1}^N \vec{X}_i^2$. Cada nó da

CF – tree contém estatísticas suficientes para descrever o conjunto de dados e toda a informação dos *CFs* nos nós que estão hierarquicamente abaixo na árvore (ZHANG; LIVNY, 1996).

Embora o algoritmo BIRCH construa a hierarquia de grupos com apenas uma leitura dos dados, para se obter um resultado satisfatório é necessário uma ou mais leituras para refinamento da hierarquia, tornando-o pouco adequado para fluxos de dados devido à restrição de apenas uma leitura.

Outro método desenvolvido para construir agrupamento hierárquico foi proposto por Tu et al. (TU et al., 2012), os autores levaram em consideração a variância e a densidade dentro de cada grupo para realizar a união entre os grupos e assim construir a hierarquia.

3.1.2 Agrupamento baseado em particionamento

Uma das estratégias mais comum na construção de agrupamentos é a baseada no particionamento dos dados (JAIN, 2010; AMINI; WAH; SABOOHI, 2014). Esse particionamento pode ser realizado de modo que os dados concernentes a um fluxo de dados S possam pertencer a mais de um agrupamento ao mesmo tempo ou a um conjunto de grupos disjuntos, de modo que um dado pode pertencer a apenas um conjunto (JAIN, 2010). Esse último tipo de particionamento de S é de interesse desse trabalho de pesquisa, é formalmente expresso pela definição 3.

Definição 3. Um conjunto de grupos disjuntos $\mathbb{C} = \{C_1, \dots, C_m\}$ de um fluxo de dados S , tal que $\bigcup_{i=1}^m C_i = S$ e $C_i \cap C_j = \{\emptyset \mid \forall i \neq j\}$.

Diversos algoritmos de agrupamento de dados baseados no particionamento de S foram propostos na literatura da área (GUHA et al., 2000; GUHA et al., 2003; AGGARWAL et al., 2003; BERINGER; HÜLLERMEIER, 2006; BIRANT; KUT, 2007; GUHA, 2009; ERIKSSON et al., 2011; FELDMAN; LANGBERG, 2011; HAHSLER; DUNHAM, 2011; ACKERMANN et al., 2012). Esses trabalhos possuem abordagens similares, visto que todos utilizam o algoritmo *k-means* de algum modo. Segundo Jain (2010) o *k-means* é um dos algoritmos de agrupamento por particionamento mais utilizados. Para encontrar os pontos de dados mais similares entre si, para então agrupá-los, o *k-means* pode utilizar, por exemplo, a medida da distância euclidiana entre os pontos de dados (SU; CHOU, 2001). Esse algoritmo é utilizado, também, para particionamento de fluxos de dados em um conjunto k de grupos (*clusters*), sendo a quantidade final de grupos k um parâmetro fornecido antes do início do processo de particionamento (GAMA, 2010). Nas seções seguintes são apresentados os algoritmos STREAM (GUHA et al., 2003) e CluStream (AGGARWAL et al., 2003), por serem base para diversos outros métodos de agrupamento por particionamento encontrados na literatura, por exemplo, (CAO et al., 2006; CHAKRABARTI; KUMAR; TOMKINS, 2006; JAIN, 2010; GABER; ZASLAVSKY; KRISHNASWAMY, 2005; FELDMAN; LANGBERG, 2011; TRAN; SATTLER, 2013; FORESTIERO; PIZZUTI; SPEZZANO, 2013; WANG et al., 2013; ZHU et al., 2014; MILLER et al., 2014).

3.1.2.1 STREAM

O STREAM foi proposto por Guha et al [Guha et al. \(2003\)](#), com o objetivo de melhorar alguns aspectos do *k-means* para contemplar o fluxo contínuo de dados. Dentre as melhorias propostas estão, principalmente, o suporte a apenas uma leitura dos dados, o uso reduzido de memória e algumas garantias de performances de execução ([GUHA et al., 2003](#)).

Esse algoritmo realiza o agrupamento de dados em duas etapas. Na primeira etapa é utilizada uma sub-rotina, chamada *local search*, para gerar os *k*-centros dos grupos. Para construir os *k*-centros essa sub-rotina executa o *k-means* sobre os dados contidos na memória, para agrupar os pontos de dados. Os *k*-centros obtidos são mantidos em memória, possibilitando, desse modo, o descarte dos dados já processados.

O STREAM atribui pesos a cada um dos *k*-centros de acordo com o número de pontos de dados contido em cada grupo, para então agrupá-los em grupos maiores na segunda etapa do algoritmo. Portanto, a segunda etapa do STREAM consiste em agrupar os *k*-centros obtidos previamente, para localizar os centros globais, construindo de maneira incremental e hierárquica uma estrutura de grupos utilizando apenas uma leitura dos pontos de dados.

Para garantir a qualidade do agrupamento, o STREAM utiliza um fator constante de aproximação, o qual indica que a resposta dada pelo STREAM é uma resposta aproximadamente ótima, obtida por meio da minimização da soma dos erros ao quadrado (*Sum Square Error*).

Em suma, a obtenção dos agrupamentos pelo algoritmo STREAM é realizada de maneira incremental com uso reduzido de memória e realizando apenas uma leitura dos pontos de dados. Entretanto esse algoritmo não leva em consideração a evolução dos agrupamentos, não dá mais valor para os dados vistos mais recentemente e é mandatário que o usuário informe a quantidade *k* de agrupamentos que serão obtidos.

3.1.2.2 CluStream

Do mesmo modo que o algoritmo STREAM ([GUHA et al., 2003](#)), visto anteriormente, o CluStream ([AGGARWAL et al., 2003](#)) também é dividido em duas etapas: uma *online* e outra *offline*. A componente *online* realiza de modo incremental as atualizações e o armazenamento do resumo das informações para serem utilizadas para construir micro grupos em diversos momentos no tempo. Essas informações são obtidas através da inferência dos pontos de dados contidos em *snapshots* (fotografias dos fluxos de dados em um determinado instante de tempo) para, na segunda etapa do algoritmo, realizar a construção de grupos maiores, de maneira *offline*, a partir dos micro grupos utilizando uma janela de tempo desejada.

O CluStream expande o vetor de características, que é a quantidade de dados no grupo, a soma desses dados e a soma ao quadrado desses dados. O vetor de características foi desenvolvido no algoritmo BIRCH ([ZHANG; LIVNY, 1996](#)), descrito na Seção 3.1.1. A expansão desse vetor de características se dá por meio do armazenamento dos microgrupos em uma estrutura

piramidal de tempo. Essa estrutura piramidal permite retornar para o usuário uma aproximação do agrupamento, baseando-se em qualquer uma das janelas de tempo definidas.

Tanto a fase *online* quanto a fase *offline*, do CluStream, utilizam o algoritmo *k-means* para realizar o agrupamento. A etapa *online* primeiramente cria um grande número de microgrupos e então, à medida que novos dados chegam, eles são processados e adicionados ao microgrupo mais próximo e as estatísticas do microgrupo hospedeiro são incrementadas, para manter atualizadas as informações dos dados agrupados. A partir desse ponto o usuário poderá iniciar a etapa *offline* do CluStream, indicando qual a *snapshot* deverá ser considerada, mediante um parâmetro h e, também, fornece a quantidade k de grupos que deverá ser retornada.

Diferentemente do algoritmo STREAM (GUHA *et al.*, 2003), no CluStream é possível acompanhar a evolução dos grupos ao longo do tempo, comparando os grupos gerados em diferentes momentos. É possível identificar quais grupos foram criados e quais deixaram de existir. Todavia, a importância dada para os dados vistos mais recentemente é a mesma que é dada para os primeiros dados analisados pelo CluStream. Outra limitação semelhante ao algoritmo STREAM (GUHA *et al.*, 2003), é a necessidade de fornecer um número pré-determinado de grupos.

3.1.3 Agrupamento baseado em grade

Outro tipo de técnica de agrupamento empregada em fluxos de dados é classificada como agrupamento baseado em uma estrutura de grade multirresolução, em inglês é conhecida como *grid-based clustering*. Essa classificação ocorre porque o espaço de dados é dividido em uma determinada quantidade de células formando uma estrutura de grade, então é executado o agrupamento dos dados de acordo com a distribuição dos dados nas grades (HAN; PEI; KAMBER, 2011; AMINI; WAH; SABOOHI, 2014). Diferentemente do que ocorre na maioria dos métodos de agrupamento por particionamento, em que é necessário informar o número de grupos que se deseja criar e que encontra grupos do tipo esfera, o agrupamento baseado em grade encontra grupos naturais de modo automático e independente do número de objetos, além das formas encontradas dos agrupamentos serem das mais variadas (HAN; PEI; KAMBER, 2011). Devido ao fato de conseguir mapear um fluxo de dados, potencialmente infinito, para uma estrutura de grade com um número limitado de células, o agrupamento baseado em grade pode se candidatar como uma escolha natural para realizar o agrupamento de fluxos de dados.

Dentre os muitos métodos baseados em grade propostos na literatura, destacam-se (AMINI; WAH; SABOOHI, 2014): o STING (WANG *et al.*, 1997), um dos primeiros métodos baseados em grade; o Wavecluster (SHEIKHOESLAMI; CHATTERJEE; ZHANG, 2000), que utiliza transformações de *wavelet* para encontrar similaridade em uma base de dados convencional, o D-Stream II (TU; CHEN, 2009) que utiliza um componente *online* para mapear cada dado para uma grade e outro componente *offline* que computa a densidade das grades para agrupá-las. Outro método baseado em uma estrutura de grade é o Fractal Clustering (FC) (BARBARA; CHEN,

2009). Esses trabalhos têm em comum o fato de realizarem o agrupamento ponto a ponto de um fluxo de dados e apenas os dois últimos procuram lidar com a característica evolutiva dos dados. Especificamente, o trabalho de Barbara e Chen (2009) foi uma das fontes de inspiração para o desenvolvimento de um dos métodos desse trabalho de doutorado, o eFCC, e por esse motivo é explicado em mais detalhes na Seção seguinte.

3.1.3.1 FC

O método proposto por Barbara e Chen (2009) busca determinar a dimensão fractal de um agrupamento de dados chamado *Fractal Clustering* (FC). A ideia geral desse método é que ao agrupar um ponto de dados em um grupo a alteração da dimensão fractal, definida na Seção 4, seja mínima. Para realizar o agrupamento de dados o FC utiliza três etapas: 1) Processo de inicialização, 2) Processo Incremental e 3) Reorganização.

Na primeira etapa os autores utilizam um algoritmo de inicialização (*bootstrap*) que dada uma amostra de dados unidimensional, associa cada um em uma grade, calcula a dimensão fractal de correlação $\mathcal{D}_2$, dada pela Fórmula 4.2, das grades e agrupa as mais similares utilizando um algoritmo como o *k-means*, fornecendo a quantidade K de grupos desejada.

Na segunda etapa, após a formação dos grupos iniciais, o FC adiciona novos pontos de dados x_i aos grupos existentes. Sendo C_k os grupos criados na primeira etapa e $FD_2(C_i)$ a dimensão fractal do grupo C_k , então para realizar a adição, o FC calcula para todos os grupos o valor na nova dimensão fractal $FD_2(C'_k), \forall k \in K$. O grupo que apresentar a menor variação $\min \| FD_2(C'_k) - FD_2(C_k) \|$ e que não extrapolar o limiar τ fornecido pelo usuário, será o grupo escolhido para receber x_i .

Na terceira etapa, o método FC irá reorganizar os agrupamentos formados realizando a divisão ou a união de agrupamentos. Um agrupamento será dividido se, a partir do momento que o grupo começou a receber dados, $FD_2(C_k)$ sofrer um grande mudança, indicando que os dados dentro do grupo já não formam um agrupamento, necessitando realizar uma divisão. Para proceder com a divisão, a primeira etapa é refeita utilizando a primeira camada da grade do agrupamento. Entretanto, após refazer essa etapa e encontrar os novos grupos bases, todos os dados pertencentes ao agrupamento deverão ser relidos do disco rígido para serem reprocessados, o que em fluxos de dados não é um processo trivial de ser realizado. Já para realizar a união de dois agrupamentos é verificado quão perto estão os dois grupos e se a proximidade deles for menor do que a proximidade interna dos grupos, então eles serão unidos.

Devido as características de trabalhar apenas com pontos de dados, operar com uma quantidade inicial fixa de grupos e ter a necessidade de realizar mais de uma leitura dos dados, para realizar a divisão de um grupo, o FC não foi considerado para ser avaliado frente aos métodos propostos nesta tese. Pois os métodos propostos para agrupar fluxos de dados, não possuem limite de construção de agrupamentos, não necessitam realizar mais de uma leitura dos dados e acompanham de maneira mais natural a evolução dos dados.

Figura 13 – Diferença do agrupamento de dados para o agrupamento de fluxos de dados.

Fonte: Elaborada pelo autor.

3.1.4 Agrupamento de Fluxo de Dados

A ideia básica subjacente do agrupamento de fluxo de dados, em inglês muitas vezes conhecido como (*Clustering Variables*), é a de encontrar grupos de fluxos de dados que tenham comportamento semelhante no mesmo intervalo de tempo (GAMA, 2010). Nesse contexto, o objetivo é particionar $\mathbb{S}$ em grupos disjuntos C em que cada fonte de fluxo de dados (por exemplo, cada sensor em uma rede) é agrupada em um conjunto disjunto de fluxos de dados com base em suas similaridades. A partir dessa descrição um particionamento de $\mathbb{S}$ é formalmente expresso pela definição 4.

Definição 4 (Agrupamento de *Data stream* unidimensional). Sendo $\mathbb{S} = \{S_1, S_2, \dots, S_N\}$ um conjunto de fluxos de dados unidimensionais. Um coleção de grupos disjuntos $\mathbb{P} = \{C_1, \dots, C_m\}$ de $\mathbb{S}$ é formado de tal modo que $\bigcup_{i=1}^m C_i = \mathbb{S}$ e $C_i \cap C_j = \{\emptyset \mid \forall i \neq j\}$.

A Figura 13 exemplifica a distinção entre agrupamento de dados e agrupamento de fluxos de dados. No agrupamento de dados, cada ponto de dados pertencente aos fluxos de dados são agrupados individualmente (Figura 13(a)). Já no agrupamento de fluxos de dados, os fluxos são agrupados em sua integralidade, com base em sua maior semelhança (Figura 13(b)).

Entretanto, nos dias de hoje os fluxos de dados podem ser multidimensionais, por exemplo, os provenientes de sensores agrometeorológicos, sendo imprescindível definir o conceito de agrupamento de fluxos de dados multidimensionais. Esse conceito é definido como:

Definição 5 (Agrupamento de *Data stream* multidimensional). Um conjunto de fluxos de dados multidimensional é formalizado como $\mathbb{S}^D = \{DS_1, DS_2, \dots, DS_N\}$. Um coleção de grupos disjuntos multidimensionais $\mathbb{C} = \{C_1, \dots, C_m\}$ de $\mathbb{S}^D$ é formado de tal modo que $\bigcup_{i=1}^m C_i = \mathbb{S}$ e $C_i \cap C_j = \emptyset$, para todo $(i \neq j)$.

Para realizar o agrupamento de fluxos de dados, em geral, os métodos propostos na literatura delimitam a quantidade de dados observada, utilizando uma janela deslizante w_{s_i} , para cada fluxo de dados s_i , e, então, realizam a sumarização dos dados w_{s_i} , para então agrupar os mais semelhantes.

Poucos trabalhos abordam agrupamentos de fluxos de dados, dentre os quais destacam-se: o ODAC - *Online Divisive-Agglomerative Clustering system* apresentado em (RODRIGUES; GAMA; PEDROSO, 2006; RODRIGUES; GAMA; PEDROSO, 2008), o método desenvolvido por Beringer e Hüllermeier (2006), o POD-Clus (*Probability and Distribution-based Clustering*) desenvolvido por Chaovalit (2012), o ECM (*Evolving Clustering Method*) desenvolvido por Widiputra, Pears e Kasabov (2011) e o TS-Stream desenvolvido por Pereira e Mello (2014). Em geral, esses métodos não tratam dados multidimensionais, com exceção do POD-Clus. Entretanto, o POD-Clus não correlaciona as dimensões para construir os agrupamentos.

Devido ao fato dos métodos mencionados acima possuírem a mesma estratégia de agrupamento que os métodos desenvolvidos nessa tese, então nas Seções seguintes cada um deles é descrito em mais detalhes.

3.1.4.1 Método de Beringuer e Hüllermeier

Um dos primeiros métodos desenvolvidos para realizar o agrupamento de fluxos de dados é o método proposto por Beringer e Hüllermeier (2006). Nesse trabalho, os autores buscaram agrupar fluxos de dados que possuíssem algum tipo de semelhança. Para identificar a semelhança entre os fluxos, os autores utilizaram uma janela deslizante W para restringir a porção de dados visualizadas e, então, realizaram uma normalização e calcularam o quão próximos estavam os fluxos de dados.

A utilização da janela deslizante teve dupla finalidade: a primeira foi restringir os dados observados e, a segunda, foi para dar mais peso aos dados vistos mais recentemente. Para isso os autores propuseram uma janela deslizante W composta pelos w mais recentes dados, sendo W particionada em m blocos de tamanho v , tal que $w = m \times v$, conforme mostrado a seguir para um fluxo de dados D .

$$D = (\underbrace{d_0, d_1, \dots, d_{v-1}}_{B_1} | \underbrace{d_v, d_{v+1}, \dots, d_{2v-1}}_{B_2} | \dots | \underbrace{d_{(m-1)v}, d_{(m-1)v+1}, \dots, d_{w-1}}_{B_m})$$

E para atribuir peso aos dados vistos mais recentemente, o algoritmo de Beringuer e Hüllermeier aplica ao ponto de dado d_i a fórmula C^{w-i-1} , sendo que $0 < C \leq 1$; w é o número de observações da janela deslizante e i a posição do ponto de dado na janela. Então, para realizar o cálculo do peso de todos os ponto de dados observados, utiliza-se o vetor de pesos $\vec{V} = (C^{w-1}, C^{w-2}, \dots, C^0)$, desse modo, o produto de $\vec{V} \odot D = (C^{w-1}d_0, C^{w-2}d_1, \dots, C^0d_{w-1})$.

A normalização é aplicada para que os dados transformados tenham média igual a 0 e desvio padrão igual a 1, obtido por meio da subtração de cada ponto de dado pela média e dividindo o resultado pelo desvio padrão. Após realizada a normalização é então calculada a distância entre os pontos de cada um dos fluxos de dados por intermédio da Transformada

Discreta de Fourier (DFT), conforme a equação abaixo:

$$DFT_f(D) = \frac{1}{\sqrt{w}} \sum_j^{w-1} d_j \times \exp\left(\frac{-i2\pi f j}{w}\right) \quad (3.1)$$

tal que, $i = \sqrt{-1}$ e $f = 0, 1, \dots, w - 1$.

Esta técnica utiliza o algoritmo *k-means* como mecanismo de agrupamento principal, obtendo uma quantidade k de grupos de modo automático, acompanhando, desta forma, a evolução dos grupos. Essa evolução é realizada por meio do aumento ou diminuição do número de grupos (k) em mais ou menos um. Para aumentar k , um novo centro do grupo é definido, escolhendo-se entre os candidatos do fluxo de dados que possuem distâncias cada vez maior para os seus centros de grupo. Os candidatos são então avaliados por uma medida de qualidade chamada de índice de separação, a fim de selecionar o fluxo que se obtenha a melhor qualidade de separação. Para diminuir k , a técnica tenta dissolver cada um dos k grupos existentes e redistribuir cada um dos membros aos grupos remanescentes. A dissolução do grupo que, posteriormente, resultar em um melhor índice de separação, é o grupo escolhido para ser removido.

Algumas das limitações do método de Beringuer e Hüllermeier são evidentes, como a necessidade de testar todos os k grupos para decidir qual retirar. Outro limitante é o uso da distância Euclidiana, que pode mascarar a proximidade de séries que não possuam alinhamento no eixo do tempo (PEREIRA, 2013).

3.1.4.2 Online Divisive Agglomerative Clustering

Desenvolvido por Rodrigues, Gama e Pedroso (2008), o ODAC (*Online Divisive Agglomerative Clustering*) é uma técnica de agrupamento hierárquico de fluxos de dados que utiliza uma árvore como estrutura hierárquica dos grupos. Cada folha da árvore criada representa um cluster com um conjunto de fluxos de dados. Já a união das folhas representa o conjunto total de fluxos de dados da hierarquia de grupos (GAMA, 2010). Esse sistema mensura a dissimilaridade entre séries temporais, sobre fluxos de dados, calculando primeiramente o coeficiente de correlação de Pearson (*corr*). Dada duas séries temporais a e b a correlação de Pearson é:

$$corr(a, b) = \frac{P - \frac{AB}{n}}{\sqrt{A_2 - \frac{A^2}{n}} \sqrt{B_2 - \frac{B^2}{n}}} \quad (3.2)$$

Sendo que $A = \sum a_i$, $B = \sum b_i$, $A_2 = \sum a_i^2$, $B_2 = \sum b_i^2$, $P = \sum a_i b_i$, podem ser facilmente atualizáveis a cada novo dado que chegar. Já a dissimilaridade entre duas séries temporais a e

b é mensurada pela métrica *Rooted Normalized One-Minus-Correlation RNOMC*, dada pela equação:

$$rnomc(a, b) = \sqrt{\frac{1 - corr(a, b)}{2}} \quad (3.3)$$

com intervalo entre $[0, 1]$. Para monitorar cada um dos grupos criados, o ODAC utiliza o diâmetro de cada grupo, que é a maior distância existente entre dois fluxos de dados s_i e s_j , tal que $i \neq j$, contidas no grupo. De posse de s_i e s_j , que representam o diâmetro do grupo, é possível realizar a quebra em dois novos grupos se determinada condição, relacionada ao diâmetro, ocorrer. Para realizar essa quebra o ODAC determina que cada uma dos fluxos de dados que representavam o diâmetro do grupo, sejam, agora, os pivôs dos novos grupos. E todos os fluxos de dados remanescentes no grupo antigo são então reassociadas a um dos dois grupos criados, de acordo com a menor distância ao pivô. Deste modo, com a quebra em dois novos grupos, cada um iniciará as suas próprias estatísticas e é garantido que nenhum deles tenha um diâmetro maior que o do grupo que os originou.

O critério de quebra de um grupo, leva à construção da árvore hierárquica e é baseado nos limites de Hoeffding (HOEFFDING, 1963). Após n observações de um valor randômico r , com intervalo R e confiança $1 - \delta$, a média real de r é, pelo menos, $\bar{r} - \varepsilon$, sendo que, $\bar{r}$ representa a média observada dos exemplos e

$$\varepsilon = \sqrt{\frac{R^2 \ln\left(\frac{1}{\delta}\right)}{2n}} \quad (3.4)$$

Deste modo, cada agrupamento c_k possuirá um valor diferente de ε , ou seja, um ε_k , pois cada agrupamento possui um número diferente de fluxos de dados. No ODAC duas distâncias são consideradas diferentes somente se a separação entre elas for maior do que ε_k . Sendo $d_1(a, b)$ o diâmetro máximo, se a diferença entre a segunda maior distância $d_2(c, d) = \max(D \setminus d_1(a, b))$ no grupo for estatisticamente significativa, de acordo com o limite de Hoeffding, então a primeira, $d_1(a, b)$, é o diâmetro real do grupo. Para o ODAC aplicar o teste de divisão duas condições devem ser satisfeitas: a primeira é quando o parâmetro de sistema τ for maior que ε_k , e a segunda é quando o diâmetro verdadeiro do agrupamento for conhecido utilizando-se a fórmula 3.5.

$$d_1 - d_2 > \varepsilon_k \Rightarrow \text{diametro}(c_k) = d_1 \quad (3.5)$$

A regra descrita acima, aciona a quebra do agrupamento quando a folha da árvore tiver recebido uma quantidade suficientes de fluxos de dados que suportem o desmembramento. Entretanto, para prevenir que a hierarquia cresça indefinidamente, os autores adicionaram uma checagem extra, em que é verificado se existe uma alta diferença entre $(d_1 - \bar{d})$ e $(\bar{d} - d_0)$, onde d_0 é a menor distância entre fluxos de dados pertencentes ao agrupamento e $\bar{d}$ é a média de todas

as distâncias do agrupamento. Então baseado-se, também, na fórmula 3.6, será permitido realizar a divisão do agrupamento. O algoritmo 1 realiza a quebra de um nó.

$$(d_1 - d_0)|d_1 + d_0 - 2\bar{d}| > \epsilon k \quad (3.6)$$

Algoritmo 1 – Algoritmo que testa a divisão de dois grupos

```

1: função TESTADIVISÃO( $c_k$ )      ▷ Retorna um boolean True ou False indicando se  $c_k$  será
   dividido ou não
2:   Seja  $d_1, d_2, d_0$  e  $\bar{d}$  as distâncias definidas nas equações 3.5 e 3.6;
3:   se  $d_1 - d_2 > \epsilon_{jk}$  ou  $\tau > \epsilon_k$  então
4:     se  $(d_1 - d_0)|(d_1 - \bar{d}) - (\bar{d} - d_0) > \epsilon_k$  então
5:       cria-se  $c_x$  e  $c_y$  tornando  $a$  e  $b$  pivos de  $c_x$  e  $c_y$ , respectivamente;
6:       para todo ponto de dado remanescente  $x_i \in c_k$  faça
7:         se  $d(x_i, a) \leq d(x_i, b)$  então
8:            $x_i \in c_x$ ;
9:         senão
10:           $x_i \in c_y$ ;
11:       fim se
12:     fim para
13:     retorna True
14:   fim se
15:   fim se
16:   retorna False
17: fim função

```

Já para realizar o reagrupamento de folhas anteriormente desmembradas, o ODAC monitora a chegada de novos fluxos de dados ao grupo e atualiza as estatísticas: se o diâmetro do grupo for maior que o do Pai, então o sistema reagrupa os grupos e reinicia as estatísticas e as folhas são removidas (RODRIGUES; GAMA; PEDROSO, 2008). O algoritmo que efetua a agregação de nós pode ser observado no algoritmo 2.

Algoritmo 2 – Algoritmo que testa a agregação de dois grupos

```

1: função TESTAAGREGACAO(Agrupamento  $c_k$ )      ▷ Retorna um boolean True ou False
   indicando se  $c_k$  será agregado ou não à estrutura  $Y$ 
2:   Atualiza dissimilaridade de  $c_k$ , se necessário, e  $\epsilon_{jk} = \max(\epsilon_j, \epsilon_k)$ ;
3:   se  $\text{diametro}(c_k) - \text{diametro}(c_j) > \epsilon_{jk}$  então
4:     corta sub-arvore  $c_j$ , tornando  $c_j$  em folha e reinicia as estatísticas;
5:     retorna True;
6:   fim se
7:   retorna False;
8: fim função

```

Mesmo adiando a quebra de um grupo até que a condição do limite de Hoeffding seja satisfeita, ainda assim o ODAC possui complexidade $O(n^2)$, em que n representa o número de séries temporais do fluxo. Outro problema encontrado no ODAC é que ele atribui o mesmo

peso para todos os dados. O ODAC leva em consideração que o modelo que originou as séries temporais possui distribuição normal ou simétrica, o que pode não representar adequadamente a série temporal, conforme apontou Pereira [Pereira e Mello \(2014\)](#) em seus estudos. Outro problema do ODAC apontado por Pereira [Pereira e Mello \(2014\)](#) é o uso da correlação de Pearson, que é capaz de detectar apenas relacionamentos lineares e é sensível a ruídos e *outliers*.

Algoritmo 3 – Algoritmo ODAC - Online Divisive Agglomerative Clustering

```

1: função ODAC( $S = (s_1, s_2, \dots, s_n)$  onde  $S$  é um conjunto de séries temporais) ▷
   Retorno é estrutura de agrupamento hierárquico de Séries Temporais  $Y$  onde as folhas
    $F = (f_1, f_2, \dots, f_n)$  representam os agrupamentos
2:   enquanto !EOF faça
3:     Lê os próximos ponto de dados do instante  $t \in S$ 
4:     Atualiza as estatísticas em  $F$ 
5:     para todo folha  $f_k$  ainda não testada faça
6:       Atualiza dissimilaridade e Hoeffding Bound  $\epsilon_k$  em cada folha
7:       se TestaAgregação( $f_k$ ) ou TestaDivisão( $f_k$ ) então
8:         Atualiza Estrutura  $Y$ 
9:       fim se
10:    fim para
11:  fim enquanto
12: fim função

```

3.1.4.3 POD-Clus

O algoritmo POD-Clus, acrônimo para *Probability and Distribution-based Clustering*, foi desenvolvido por Chaovalit [\(2012\)](#). Esse algoritmo trata quatro abordagens de agrupamento: por meio de exemplos sem evolução dos grupos; por meio de exemplos com acompanhamento da evolução dos grupos; de fluxos de dados sem evolução dos grupos e, por último, de fluxos de dados com acompanhamento da evolução dos grupos. Desses quatro tipos, apenas o último é de interesse no contexto dessa tese, por se tratar de agrupamento de fluxos de dados que acompanha a evolução dos grupos, como os métodos propostos o fazem.

O POD-Clus busca manter resumos e descartar informações detalhadas dos pontos de dados, sendo que, o resumo das informações é mantido por distribuições normais. O PODClus constrói dois tipos de grupos, os grupos chamados de *outliers* e os grupos normais. Para construir seus grupos o POD-Clus utiliza três parâmetros: o primeiro é o parâmetro que determina a quantidade inicial de grupos normais k , o segundo um limiar de desvio padrão para informar qual a distância máxima necessária para agrupar um fluxo de dados com o grupo mais próximo e, o terceiro é o parâmetro que indica qual a quantidade mínima de fluxos de dados em um grupo de *outliers* é necessária para o grupo passar a ser considerado um grupo normal. Cada grupo k do POD-Clus recebe n pontos de dados, sendo que a quantidade de pontos que cada grupo recebe é muito maior que a quantidade de grupos, resultando em uma complexidade de processamento de $O(kn)$, e cada um dos grupos tem as estatísticas de média μ , desvio padrão σ e a matriz de

covariância atualizadas sempre que novos dados chegam. Essas estatísticas são utilizadas para mensurar a similaridade entre os fluxos de dados considerando cada atributo, de acordo com a Equação 3.7. Sendo a distância entre fluxos de dados multidimensionais DS para o centroide do grupo C definida como:

$$Dist_{DSxC} = \sum_{f=1}^F \frac{(\mu_{DS_f} - \mu_{C_f})^2}{\sigma_{DS_f}^2}, \quad (3.7)$$

Onde DS representa o fluxo de dados, C representa o grupo, f a dimensão, μ_{DS_f} a média da dimensão f no fluxo de dados DS , σ_{DS_f} o desvio padrão da dimensão f no fluxo de dados DS e μ_{C_f} é a média da dimensão f do grupo C .

O algoritmo proposto por Chaovalit (2012) também busca acompanhar a evolução de cada agrupamento. Buscando identificar o aparecimento de novos grupos, o desaparecimento de um grupo que existia, a união e a divisão de grupos. Para detectar o aparecimento de um novo grupo, quando um dos grupos de *outliers* atinge a quantidade mínima de trinta fluxos de dados, esse grupo torna-se um novo grupo pertencente ao agrupamento. Já para um grupo desaparecer o mesmo deve ficar sem receber novos dados por uma certa quantidade de tempo, pois dados antigos recebem menos importância aplicando-se um fator de enfraquecimento para eles. Para um grupo unir-se com outro, é verificado a quantidade de dados que podem estar em ambos os grupos, gerando *overlapping* entre eles. Se a quantidade de dados em *overlapping* for maior que um determinado limite, então os dois grupos são unidos. Para dividir um grupo o POD-Clus monitora sua densidade e a compara com a distribuição normal e, se houver significativa diferença, o grupo é então dividido.

Por ser baseado apenas em medidas básicas, como média, desvio padrão, a matriz de covariância e regressão linear, tornam o POD-Clus limitado quanto à descrição dos grupos formados. Além disso, supor que toda a série temporal é oriunda de uma distribuição normal também não garante uma boa representatividade dos grupos (PEREIRA, 2013).

3.1.4.4 ECM

O método *Evolving Clustering Method* (ECM) foi proposto por Song e Kasabov (2001) e adaptado por Widiputra, Pears e Kasabov (2011) para realizar a predição de várias séries temporais. Esse método constrói modelos locais em dois passos principais, que são a extração dos perfis de relacionamentos entre séries temporais e a detecção e agrupamento de tendências recorrentes na série temporal quando um determinado perfil emerge.

O ECM calcula os perfis de relacionamento entre as séries temporais utilizando a correlação de Pearson, extraíndo apenas os coeficientes mais significantes, obtidos mediante testes de significância estatística. Então, o ECM calcula a dissimilaridade entre as séries temporais por meio da equação RNOMC, desenvolvida no ODAC (RODRIGUES; GAMA; PEDROSO, 2008) e apresentada na equação 3.3. Entretanto, diferentemente do ODAC, o ECM necessita que toda a série temporal esteja disponível previamente para realizar os cálculos de modo *offline*.

Baseando-se em um conjunto de regras de decisão, o algoritmo decide como agrupar as séries optando por criar, remover ou unir grupos (PEREIRA, 2013), conforme pode ser identificado no algoritmo 4.

Algoritmo 4 – ECM

```

1: função ECM( $D = (d_1, d_2, \dots, d_n)$ )  $\triangleright$  onde  $D$  é um conjunto de séries temporais  $\triangleright$  Retorna
   agrupamentos de Séries Temporais
2:   Calcular as correlações entre D utilizando RNOMC (eq: 3.3)
3:   para  $d_1$  até  $d_n$  faça
4:     se ( $d_i, d_j$  forem correlacionados) e ( $d_i, d_j$  não pertencerem a nenhum grupo) então
5:       cria um novo agrupamento e armazena  $d_i, d_j$ 
6:     fim se
7:     se ( $d_i, d_j$  forem correlacionados) e ( $d_i \in$  algum grupo) então
8:       se  $d_j$  for correlacionado  $\forall d_i$  então
9:         armazena  $d_j$  ao grupo de  $d_i$ 
10:      senão se (correlação  $d_i, d_j >$  maior correlação dos membros de  $d_i$ ) e ( $d_j$  não é
        correlacionando a nenhum membro de  $d_i$ ) então
11:        remove  $d_i$  do grupo e aloca  $d_i$  e  $d_j$  a um novo grupo
12:      fim se
13:    fim se
14:    se ( $d_i, d_j$  forem correlacionados) e ( $d_i, d_j \in$  algum grupo) então
15:      se ( $d_i$  for correlacionado  $\forall d_j$ ) e ( $d_j$  for correlacionado  $\forall d_i$ ) então
16:         $\cup(d_i, d_j)$ 
17:      senão se (correlação  $d_i, d_j >$  correlação  $d_j$  com os membros do próprio  $d_j$ ) e ( $d_j$ 
        é correlacionado  $\forall$  grupo de  $d_i$ ) então
18:        remove  $d_j$  do seu grupo
19:        aloca  $d_j$  ao grupo de  $d_i$ 
20:      senão se (correlação  $d_i, d_j >$  maior correlação  $d_i, d_j$  com seus respectivos mem-
        bros) e ( $d_i$  não for correlacionado com nenhum membro de  $d_j$ ) e ( $d_j$  não for correlacionado
        com nenhum membro de  $d_i$ ) então
21:        remove  $d_i, d_j$  de seus grupos
22:        cria um novo grupo para alocar  $d_i, d_j$ 
23:      fim se
24:    fim se
25:  fim para
26: fim função

```

Devido ao fato do algoritmo utilizar a mesma medida de dissimilaridade do ODAC (RODRIGUES; GAMA; PEDROSO, 2008), ele apenas detecta relacionamentos lineares. Outro ponto que limita a aplicação do ECM é o modo como o algoritmo decide pela união, criação ou adição de um novo ponto de dado ao agrupamento, por exemplo, nas linhas 6 à 8 do ECM (4), o ponto de dado d_j somente será agregado ao agrupamento de d_i se d_j for correlacionado com todos os pontos de dados existentes em d_i , portanto, se d_i possuir mil pontos de dados e d_j não for correlacionado a apenas um desses ponto de dados, então d_j não será agrupado a d_i .

3.1.4.5 TS-Stream

O TS-Stream foi desenvolvido por [Pereira e Mello \(2014\)](#), para realizar o agrupamento de séries temporais utilizando uma abordagem alternativa aos métodos vistos anteriormente que utilizam a correlação de Pearson para determinar a similaridade entre as séries temporais. O TS-Stream utiliza medidas descritivas para realizar a caracterização do modelo gerador, das séries temporais, de modo mais completo e buscando uma melhor qualidade nos agrupamentos finais.

O autor utilizou como principal critério de validação o *Adjusted Rand Index* (ARI), que se baseia na contagem de pares de objetos, ou no caso do TS-Stream nos pares de séries temporais, nos quais duas partições concordam e duas discordam, tal que, qualquer par de objetos pode ser colocado em uma de quatro categorias: i) eles estão no mesmo grupo em ambas as partições (N_11), ii) eles estão em grupos diferentes em ambas as partições (N_00), iii) eles estão no mesmo grupo na primeira partição, mas em grupos diferentes na segunda (N_10) e iv) eles estão em grupos diferentes na primeira partição, mas no mesmo grupo na segunda (N_01).

Dadas duas partições de um conjunto S de M objetos, seja $U = U_1, U_2, \dots, U_R$ a primeira partição com R grupos e $V = V_1, V_2, \dots, V_C$ a segunda partição com C grupos. Seja também $\bigcap_{i=1}^R U_i = \bigcap_{j=1}^C V_j = \emptyset, \bigcup_{i=1}^R U_i = \bigcup_{j=1}^C V_j = S$. É possível criar uma matriz de contingência $R \times C$ na qual cada ponto de dado n_{ij} indica o número de objetos em comum aos grupos U_i e V_j .

O *Adjusted Rand Index* tem valor esperado 0 para duas partições aleatórias e 1 para partições idênticas. Deste modo o ARI pode ser calculado conforme descrito na Equação 3.8 abaixo, na qual n_{ij} é um elemento da matriz de contingência, a_i é a soma da i -ésima linha e b_j a soma da j -ésima coluna.

$$ARI = \frac{\sum_{i,j} \binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2} \right] / \binom{N}{2}}{\frac{1}{2} \left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2} \right] - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2} \right] / \binom{N}{2}} \quad (3.8)$$

O TS-Stream constrói uma árvore hierárquica de maneira *top-down*, em que inicialmente, todas as séries temporais se encontram na raiz da árvore e, então, realiza sucessivas divisões e agregações para formar a árvore. Cada nó intermediário realiza um teste binário do tipo $valor_{atributo} \leq x$ sobre determinado atributo calculado por meio de medidas descritivas, dentre as quais o expoente de Hurst e os modelos paramétricos de Box e Jenkins ([PEREIRA, 2013](#)). Ao atingir um nó folha, a série é então agrupada junto às demais séries já agrupadas na mesma folha.

O algoritmo principal do TS-Stream recebe como entrada um conjunto de fluxos de dados $D = d_1, d_2, \dots, d_m$ de tamanho infinito e três parâmetros α, β e λ , que são utilizados para determinar o grupo ou a divisão dos nós folha. Na primeira etapa o TS-Stream realizará o cálculo das medidas descritivas, para cada fluxo d_i , gerando uma matriz $m \times n$, em que m é a quantidade de fluxos de dados e n a quantidade de atributos obtidos pelos cálculos das medidas descritivas.

Na segunda etapa a matriz é normalizada utilizando o z -score : $x = (x - \mu)/\sigma$, tal que μ representa a média e σ o desvio padrão. Na terceira etapa o TS-Stream busca o atributo mais significativo, por meio da maximização do ganho da variância ponderada, para dividir o nó raiz em dois nós filhos, que melhor representem o modelo gerador de cada um. Então o TS-Stream passa a fase de agrupamento dos fluxos de acordo com a estrutura atual da árvore, iniciando a fase de atualização do modelo, na qual é avaliada a necessidade de agregação ou divisão de folhas, realizando esse procedimento de maneira recursiva.

Embora o TS-Stream atinja uma qualidade melhor dos agrupamentos frente ao ODAC (RODRIGUES; GAMA; PEDROSO, 2008), segundo os experimentos realizados pelo autor, é possível observar algumas restrições, como a não realização incremental dos cálculos das medidas descritivas, que é uma das grandes vantagens do ODAC. Outro limitante é o uso de parâmetros para determinar pela agregação ou divisão de nós folhas. Também é um limitante a complexidade do método, que é da ordem de $O(n^2)$.

3.2 Medidas de Qualidade de Agrupamentos

A validação dos agrupamentos é uma importante tarefa em mineração de dados (LIU *et al.*, 2010) e de extrema relevância no agrupamento de fluxos de dados para mensurar a qualidade dos grupos obtidos entre os diferentes métodos (LIU *et al.*, 2010; CHAOVALIT, 2012). Frequentemente não há nenhum rótulo associado ao fluxo de dados. Desse modo, validar os grupos formados e identificar se os fluxos de dados foram apropriadamente agrupados não é uma tarefa trivial (ARBELAITZ *et al.*, 2013; GAMA, 2010; AGGARWAL, 2007). A validação dos diversos métodos de agrupamento pode transformar-se em uma atividade realmente desafiadora, basicamente porque não há nenhuma informação rotulada nos dados que indique de que os fluxos de dados foram agrupados no grupo correto, nem alguma informação que possa servir de base para comparação entre os métodos (ARBELAITZ *et al.*, 2013; CHAOVALIT, 2009; LIU *et al.*, 2010). Por esse motivo, muitas abordagens de validação de agrupamentos foram propostas na literatura, em relação a disponibilidade de rótulos de classes correto (BRUN *et al.*, 2007; AMIGÓ *et al.*, 2009; LIU *et al.*, 2010).

Técnicas de agrupamento de dados são avaliadas pela qualidade dos grupos que elas produzem e podem ser categorizadas em dois grupos, dependendo se existe ou não um conjunto de validação (*ground-truth*), que são: medidas (ou índices) internas ou externas (BRUN *et al.*, 2007; HAN; PEI; KAMBER, 2011). Os índices de qualidade internos são utilizados quando não há um conjunto de validação (*ground-truth*) que indica a correta clusterização dos dados. Esse tipo de índice frequentemente é a única opção para validar métodos de agrupamento de fluxos de dados. Os índices de qualidade externos são utilizados quando existe informações de rótulo dos dados. Ou seja, quando há informação do tipo de classe ou agrupamento ao qual o dado pertence ou, quando há um conjunto de validação.

Entre todos os critérios de validação, alguns índices, como o Coeficiente de Silhueta e o Soma dos Erros Quadrados (SSE - Sum Square Error), são comumente utilizados para a avaliação de agrupamento de fluxos de dados (CHAOVALIT, 2012).

Contudo, diversos índices de validação foram desenvolvidos na área de agrupamento de dados. Esses índices, buscaram acessar a qualidade dos agrupamentos formados por diferentes tipos de dados (numéricos ou categóricos), várias dimensionalidades e pelo tipo de agrupamento (ponto a ponto ou do *data stream*). Na Seção 3.2.1 são apresentados os índices de validação interna. E a Seção 3.2.2 apresenta alguns índices de validação externa.

3.2.1 Índices Internos

Os índices internos mensuram a qualidade dos grupos obtidos utilizando apenas o resultado alcançado pela técnica de agrupamento utilizada, ou seja, utiliza-se apenas os próprios grupos resultantes. Se algum tipo de informação sobre a correta formação dos agrupamentos estivesse disponível, não haveria necessidade de utilizar esse tipo de índice. Pois resultados mais acurados seriam obtidos utilizando-se índices externos. Todavia, os índices internos são comumente utilizados na avaliação de agrupamentos de fluxos de dados devido a dificuldade de constantemente rotular os fluxos de dados potencialmente infinito e que podem ser gerados a cada milésimo de segundos.

O uso dos índices internos ocorre tendo-se como premissa que os grupos formados deverão possuir dados que são mais próximos entre si, do que dados que estão mais distantes (CHAOVALIT, 2012). Com base nessa premissa, dois métodos de avaliação de agrupamentos são frequentemente utilizados o método SSE e o Coeficiente de Silhueta. Esses dois índices e o índice de Dunn são apresentados a seguir.

3.2.1.1 Índice SSE

O índice SSE mensura o total da soma dos erro ao quadrado das distância de um determinado ponto em relação ao centroide de seu próprio grupo (BRUN *et al.*, 2007). Sendo K o número total de agrupamento, $k \in K$ um grupo do total de agrupamentos, x_i um ponto de dado pertencente a um *data stream* e, μ_k o centroide do agrupamento k , então o SSE é definido como:

$$SSE = \sum_{k=1}^K \sum_{i \in k} (x_i - \mu_k)^2 \quad (3.9)$$

A fórmula 3.9 pode ser facilmente estendida para medir *data stream* multidimensional D , onde $1 \leq d \leq D$, sendo x_i^d um ponto no espaço multidimensional D e μ_k^d o centroide do agrupamento k na dimensão d , então a Soma dos Erros ao Quadrado Multidimensional (SSE^D) é dado pela fórmula:

$$SSE^D = \sum_{k=1}^K \sum_{d=1}^D \sum_{i \in k} (x_i^d - \mu_k^d)^2 \quad (3.10)$$

Quanto menor o valor do índice SSE menor é a distância dos pontos de dados para os centroides, consequentemente, assume-se que há uma boa formação de agrupamentos. Todavia, o SSE e SSE^D não são um bom critério de avaliação quando se está mensurando a qualidade de agrupamentos com formação arbitrária, pois esse tipo de formação não fornece valores mínimos de SSE ou SSE^D .

3.2.1.2 Coeficiente de Silhueta

O Coeficiente de Silhueta (CS), ou apenas Silhueta, mede o quão próximos (coesos) estão os dados um do outro em um agrupamento (BRUN *et al.*, 2007; LIU *et al.*, 2010; ARBELAITZ *et al.*, 2013). Para definir a Silhueta, considere um objeto x_i pertencente ao agrupamento A . A média da distância de x_i para todos os outros objetos dentro do mesmo agrupamento A é dada por a_i . Essa medida é também conhecida como distância **intragrupo**. Considerando o agrupamento $K \neq A$, a média da distância de x_i para todos os pontos de dados pertencentes ao agrupamentos k , é dada por $dist(x_i, K)$, então, faz-se $b_i = \min(dist(x_i, K))$. Essa medida é também conhecida como medida **intergrupos**. Desse modo, a Silhueta é calculada como:

$$CS(x_i) = \frac{b_i - a_i}{\max(b_i, a_i)} \quad (3.11)$$

Os valores de Silhueta variam de -1 até 1, onde valores próximos a 1 indicam uma boa escolha de agrupamento e valores próximos a -1 indicam que o objeto x_i não foi agrupado no grupo correto. Para verificar a boa formação dos grupos, utiliza-se a média dos valores dos pontos de dados x_i , tal que $1 \leq i \leq N$.

Entretanto, é fácil perceber que o custo de calcular CS é quadrático, pois todos os pontos devem ser verificados contra todos os pontos. No trabalho proposto em (HRUSCHKA; CAMPELLO; CASTRO, 2004) os autores propuseram o Coeficiente da Silhueta Simplificada, que consiste em calcular a_i com relação ao centroide de seu próprio grupo. Do mesmo modo, os autores propuseram calcular $dist(x_i, K)$ em relação ao centroide do grupo k . Com essas modificações o custo computacional passou a ser linear. Devido a essa melhoria, os resultados dessa tese utilizarão o Coeficiente da Silhueta Simplificada.

3.2.1.3 Índice de Dunn

Similar ao Coeficiente de Silhueta, o índice Dunn também mede quão coeso está a formação dos agrupamentos. Sendo, $d_{\min}$ a distância mínima entre pontos de dados de agrupamentos diferentes e $d_{\max}$ a maior distância entre pontos de um mesmo agrupamento (diâmetro). O índice de Dunn é a proporção da menor distância entre agrupamentos diferentes e o maior diâmetro entre os agrupamentos. Dado dois agrupamentos $k, l \in K$, sendo x_i os pontos de dados pertencentes a k e x_j os pontos de dados pertencentes a l , então a distância entre os agrupamentos k e l é dada por:

$$dist(k, l) = \min \|x_i^k - x_j^l\| \quad (3.12)$$

Então, $d_{min} = \min dist(k, l)$. Para cada agrupamento $k \in K$, o diâmetro entre dois pontos i, j pertencentes a K , é dado por:

$$diam(k) = \max \|x_i - x_j\| \quad (3.13)$$

Então, d_{max} é a maior distância entre K , dada por:

$$d_{max} = \max_{1 \leq k \leq K} diam(k) \quad (3.14)$$

Então o índice de Dunn é dado por:

$$Dunn = \frac{d_{min}}{d_{max}} \quad (3.15)$$

Um alto valor do índice Dunn sugere que os agrupamentos estão mais compactos e distantes um dos outros, pois a distância entre os grupos tende a ser maior e o diâmetro deve ser menor.

3.2.2 Índices Externos

Os índices externos mensuram a qualidade dos agrupamentos formados quando há informação de qual agrupamento os pontos de dados - ou os fluxos de dados, que é o foco desta tese - deveriam pertencer. Essa informação, que indica a correta associação dos fluxos de dados aos seus respectivos agrupamento, é chamada de *ground-truth*. Então, a qualidade dos agrupamentos é mensurada comparando-se o resultado obtido, pelas técnicas de agrupamento, com a informação de *ground-truth* fornecida.

Diversos índices externos foram propostos na literatura, com o objetivo de avaliar a qualidade dos agrupamentos obtidos. Nessa Seção serão revisados os índices Precisão e Revocação. Todavia, antes de exemplificar os métodos, é necessário definir a terminologia envolvida e que será utilizada em ambos os métodos. Uma lista mais detalhada de índices externos e das terminologias envolvidas pode ser encontrada em (HRUSCHKA; CAMPELLO; CASTRO, 2004; BRUN *et al.*, 2007; AMIGÓ *et al.*, 2009; HRUSCHKA *et al.*, 2009; LIU *et al.*, 2010; ARBELAIZ *et al.*, 2013; AMINI; WAH; SABOOHI, 2014).

3.2.2.1 Terminologia empregada em índices externos

Para melhor entender como os índices externos funcionam, é necessário, primeiramente, definir alguns conceitos associados a correta inclusão ou não de um ponto de dados x_i ao seu agrupamento. Considerando o conjunto de agrupamentos A^m como sendo o resultado obtido pelo método que se está avaliando, A^c o conjunto de agrupamentos fornecido como sendo os corretos agrupamentos *ground-truth* e x_i um ponto de dado (ou fluxo de dados), então algumas situações podem ocorrer:

- **Verdadeiro Positivo (VP):** x_i está agrupado no mesmo grupo em A^m e A^c ;

- **Falso Positivo (FP):** x_i esta agrupado no grupo k em A^m mas não em k em A^c ;
- **Falso Negativo (FN):** x_i não esta agrupado no grupo k em A^m mas esta agrupado em k em A^c ;
- **Verdadeiro Negativo (VN):** x_i não esta agrupado no grupo k em A^m e, também, não esta agrupado em k em A^c ;

As situações apresentas acima podem ser representada graficamente por meio de uma Matriz de Confusão, conforme mostra a Figura 14.

Figura 14 – Representação dos conceitos empregados em índices externos por meio de uma Matriz de Confusão.

		Fornecido no ground-truth	
		Positivo	Negativo
Agrupado pelo Método	Positivo	VP	FP
	Negativo	FN	VN

Fonte: Elaborada pelo autor.

3.2.2.2 Índice Precisão

O Índice de Precisão mensura qual a porcentagem dos dados (ou dos fluxos de dados) foram corretamente agrupadas em cada grupo em relação a quantidade total de dados associados ao grupo. Desse modo, a precisão é dada pela fórmula:

$$Precisao = \frac{VP}{VP + FP} \quad (3.16)$$

3.2.2.3 Índice Revocação

A Revocação mensura qual a porcentagem dos dados (ou dos fluxos de dados) foram corretamente agrupadas em cada grupo em relação a quantidade total real de dados que deveria ter sido associada ao grupo.

$$Revocao = \frac{VP}{VP + FN} \quad (3.17)$$

3.3 Considerações finais

Conforme apresentando em cada um dos métodos de agrupamento, discutidos neste capítulo, possuem características favoráveis e pontos que precisam de melhorias. Essas limitações

incluem: a necessidade de parâmetros de percepção dos especialistas da área, não suportar dados multidimensionais e, o uso de medidas descritivas para representar os fluxos de dados. A complexidade é outro problema encontrado nos métodos, sendo a maioria deles quadráticos em relação ao número de dados lidos, inviabilizando o uso com fluxos de dados potencialmente infinitos. Há problemas, também, na representatividade do conhecimento descoberto, de modo que muitas vezes os resultados não são satisfatórios. Alguns métodos necessitam, ainda, que todos os dados estejam disponíveis no início do processamento e, outros, calculam a distância entre os pontos de dados estaticamente e dão a mesma importância a dados antigos e novos. Ou ainda, indicam a formação de grupos anômalos apenas pelo número de fluxos de dados agrupados. O Quadro 1 resume as principais características os métodos apresentados anteriormente.

Quadro 1 – Quadro comparativo entre os métodos apresentados

Método	Técnica de Agrupamento	Tipo de Fluxo de dados	Tipo de medida	Detecta Outlier	Quantidade de leituras
STREAM	Pontos de Dados	Unidimensional	Descritiva	Não	1
CluStream	Pontos de Dados	Unidimensional	Descritiva	Não	1
FC	Pontos de Dados	Multidimensional	Fractais	Sim	>1
Beringuer e Hüllermeier	Fluxo de dados	Unidimensional	Descritiva	Não	1
ODAC	Fluxo de dados	Unidimensional	Descritiva	Sim	1
POD-Clus	Fluxo de dados	Multidimensional	Descritiva	Sim	1
ECM	Fluxo de dados	Unidimensional	Descritiva	Sim	1
TS-Stream	Fluxo de dados	Unidimensional	Descritiva	Sim	1

Como é possível observar, dos métodos que agrupam fluxos de dados, apenas o POD-Clus trata fluxos de dados multidimensionais. Todavia, o POD-Clus desmembra as dimensões e as trata de modo independente. Outra característica interessante dos métodos apresentados que agrupam fluxos de dados é a de que eles utilizam-se de medidas descritivas para verificar a similaridade entre os fluxos de dados, podendo não representar o comportamento dos dados de maneira adequada. Sendo que, apenas o FC, que agrupa pontos de dados, não se utiliza de medidas descritivas para realizar o agrupamento, utilizando fractais para tal tarefa, que se aplicados a fluxos de dados podem representar melhor o comportamento dos dados.

Desse modo, conceitos de fractais, relevantes para o trabalho desenvolvido nesse doutoramento, estão inseridos no próximo Capítulo 4. E, devido as limitações mencionadas nesse capítulo, dois novos métodos são propostos no Capítulo 5.

TEORIA DE FRACTAIS

A Teoria de Fractais surgiu devido à dificuldade encontrada pela Geometria Euclidiana em classificar formas geométricas com alto grau de similaridade. Essa dificuldade decorreu principalmente porque os fractais possuem características fundamentais que os definem e os distinguem de outras formas geométricas básicas, como a autossimilaridade em diferentes níveis de escala, a complexidade infinita de representação ([SCHROEDER, 1991](#)) e uma dimensionalidade fracionária, chamada de dimensão fractal ([MANDELBROT, 1982](#)).

Devido a características tão peculiares, conceitos da Teoria de Fractais, vêm sendo empregados em diversos campos de pesquisas, como em Biomedicina ([AZEMIN *et al.*, 2016](#)), Geociências ([AI *et al.*, 2014](#)), e em Ciência da Computação, principalmente na área de Mineração de Dados ([FALOUTSOS; KAMEL, 1994](#); [ZHANG *et al.*, 2016](#); [GUO, 2017](#)). Além disso, a Teoria dos Fractais define um conjunto de técnicas que permitem identificar o nível de correlação de um conjunto de dados ([SOUSA *et al.*, 2007b](#)) ou mesmo de fluxos de dados ([SOUSA *et al.*, 2007a](#)). Na presente tese, estas técnicas foram aplicadas para mensurar correlações não lineares existentes entre os atributos de fluxos de dados de dados climáticos.

A Seção 4.1 apresenta a definição básica de fractais, assim como alguns exemplos desse tipo de estrutura. A Seção 4.2 descreve os aspectos teóricos relacionados ao cálculo da dimensão de correlação fractal de conjuntos, enquanto que na Seção 4.3 são apresentados métodos para estimar a dimensão fractal de fluxos de dados. Por fim, a Seção 4.4 sumariza as discussões do capítulo.

4.1 Definição

Fractais são caracterizados como objetos autossimilares, isto é, objetos que apresentam, a grosso modo, as mesmas características quando analisado nas mais variadas escalas e tamanhos. Assim sendo, partes do fractal - seja ele uma estrutura, um objeto ou um conjunto de dados - são

Figura 15 – Fractais encontrados na natureza

(a) Folha da samambaia

(b) Brócolis

(c) Floco de neve

Fonte: Elaborada pelo autor.

similares, exata ou estatisticamente, ao fractal como um todo (SCHROEDER, 1991).

Diversos tipos de fractais podem ser encontrados na natureza. Por exemplo, as ramificações das folhas de uma samambaia são representações de menor tamanho da folha como um todo. Já as pequenas estruturas que formam o brócolis são autossimilares ao todo deste vegetal. Um floco de neve também é outro objeto autossimilar, apresentando simetria radial. Estas três estruturas são ilustradas nas Figuras 15a, 15b e 15c, respectivamente.

Além destes, várias estruturas matemáticas são fractais. Por exemplo, o Triângulo de Sierpinski, apresentado na Figura 16a, é obtido através de diversas iterações onde, a partir de um triângulo equilátero inicial de lados ABC, retira-se um triângulo central A'B'C', formado pela mediana dos três lados originais. Para cada um dos três triângulos restantes (AB'C', A'B'C e A'BC'), retira-se novamente outro triângulo central, e assim sucessivamente.

Já o Conjunto de Mandelbrot, ilustrado pela Figura 16b, é um fractal estatisticamente autossimilar, definido como o conjunto de pontos c do espaço complexo onde a função iterativa $z_{i+1} = z_i^2 + c$, sendo $z_0 = 0$, não diverge. Outros exemplos são as Curvas de Koch (Figura 16c) e o Conjunto de Cantor (Figura 16d), que são gerados por regras geométricas bem definidas e formam objetos exatamente autossimilares.

Os fractais possuem características incomuns, e até mesmo controversas. O Triângulo de Sierpinski, por exemplo, possui perímetro infinito, proporcional a $\lim_{i \rightarrow \infty} \left(\frac{3}{2}\right)^i$. Por sua vez, a área desse fractal é estimada em $\lim_{i \rightarrow \infty} \left(\frac{3}{4}\right)^i$. Como pode ser percebido, a cada iteração, o perímetro do Triângulo de Sierpinski tende a aumentar, enquanto que a sua área diminui, tendendo a nulidade. Portanto, esse fractal não obedece às propriedades de um Espaço Euclidiano nem unidimensional, visto que seu perímetro não é um valor finito, e nem multidimensional, uma vez que não possui área. Dessa forma, é razoável considerar que o Triângulo de Sierpinski possui um valor de dimensionalidade fracionário, que é chamado de *Dimensão Fractal* (MANDELBROT, 1982) e apresentado na próxima Seção 4.2 a seguir.

Figura 16 – Fractais definidos matematicamente

Fonte: Elaborada pelo autor.

4.2 Dimensão Fractal

Contrariamente à Geometria Euclidiana, onde o valor da dimensão representa a dimensionalidade do espaço em que dado objeto está inserido, a Dimensão Fractal $\mathcal{D}$ pode ser considerada como uma representação do nível de irregularidade que um fractal possui (MANDELBROT, 1982).

A dimensão fractal mensura o valor fractal de fractais *exatamente autossimilares*. Um fractal exatamente autossimilar é composto por M réplicas em miniatura dele mesmo, sendo cada réplica uma versão em escala reduzida $1 : s$ do fractal original (SOUSA *et al.*, 2007b). Desse modo, a dimensão fractal $\mathcal{D}$ é dada pela Definição 6 (BELUSSI; FALOUTSOS, 1995).

Definição 6 (Dimensão Fractal). Seja M o número de réplicas de um objeto autossimilar e s o fator de escala no qual cada réplica encontra-se reduzida, a Dimensão Fractal $\mathcal{D}$ é dada pela equação 4.1.

$$\mathcal{D} \equiv \frac{\log M}{\log s} \quad (4.1)$$

Como exemplo, a cada iteração do processo construtivo do Triângulo de Sierpinski, 3 novas réplicas são geradas em escala 1:2 (os lados de cada novo triângulo são formados pela metade do anterior). Assim, a dimensão fractal desse objeto é dada por $\mathcal{D} = \frac{\log 3}{\log 2} \approx 1,58$. De

modo análogo, como fractal exatamente similar, as Curvas de Koch são formadas por 4 réplicas a cada iteração, sendo que cada é gerada em escala 1:3, o que leva a uma dimensão fractal dada por $\mathcal{D} = \frac{\log 4}{\log 3} \approx 1,26$. Já o Conjunto de Cantor possui dimensão fractal $\mathcal{D} = \frac{\log 2}{\log 3} \approx 0,63$, sendo que cada iteração produz 2 réplicas em escala 1:3.

Esta definição da dimensão fractal $\mathcal{D}$ é adequada para fractais matemáticos exatamente autossimilares, com regras de construção recursivas bem definidas. Todavia, existem conjuntos de fractais para os quais não estão associadas regras de construção bem definidas, como é o caso das bases de dados e dos fluxos de dados. Estes fractais são ditos *estatisticamente autossimilares*. Para esses objetos, uma das formas mais clássicas de cálculo da dimensão fractal é através da expressão da Dimensão Fractal de Correlação $\mathcal{D}_2$, dada pela Definição 7 (BELUSSI; FALOUTSOS, 1995).

Definição 7 (Dimensão Fractal de Correlação). Dado um conjunto de dados $\mathbb{D}$ com uma quantidade de pontos finita, um intervalo de escalas $[r_1, r_2]$ no qual $\mathbb{D}$ seja estatisticamente autossimilar, e um reticulado hiper-cúbico $\mathbb{L}$ de lado $r \in [r_1, r_2]$ no qual $\mathbb{D}$ está imerso, a Dimensão Fractal de Correlação $\mathcal{D}_2$ é dada pela equação 4.2, onde $C_{i,r}$ é o valor da contagem de pontos da i -ésima hiper-célula.

$$\mathcal{D}_2 \equiv \frac{\partial \log \left(\sum_i C_{i,r}^2 \right)}{\partial \log(r)} \quad (4.2)$$

A Dimensão Fractal de Correlação $\mathcal{D}_2$ é particularmente útil para a análise de dados, pois ela pode ser utilizada para estimar a dimensão intrínseca de uma base de dados real que exiba comportamento fractal, isto é, que seja uma base de dados estatisticamente ou exatamente autossimilar (BELUSSI; FALOUTSOS, 1995). A Dimensão Fractal de Correlação $\mathcal{D}_2$ mensura o comportamento não uniforme de dados reais considerando tanto atributos (ou mesmo dimensões) correlacionados linearmente quanto aqueles com correlação não linear. Sendo assim, a $\mathcal{D}_2$ representa a dimensionalidade da base de dados com relação a dimensão E do espaço definido pelos atributos da base de dados. Por exemplo, o conjunto de pontos no espaço Euclidiano que define uma reta $y = a + bx$, com duas dimensões $[X; Y]$ (desse modo $E = 2$) possui Dimensão Fractal de Correlação $\mathcal{D}_2 = 1$, pois existe uma correlação linear entre os atributos da reta (SOUSA *et al.*, 2007b). Se este mesmo conjunto de pontos que compõe a reta for disposto em um espaço de maior dimensionalidade, como um espaço de três dimensões $[X; Y; Z]$ ($E = 3$), tal que a reta é definida por $z = a + bx + cy$, o valor de $\mathcal{D}_2$ continua sendo 1, como pode ser observado nas Figuras 17a e 17b, visto que correlação entre os pontos da reta permanece linear.

Para conjuntos de fractais estatisticamente autossimilares, a estimativa do valor da Dimensão Fractal de Correlação $\mathcal{D}_2$ pode ser obtida através do assim chamado método de *Box-counting*.

Figura 17 – Reta em espaços de dimensões 2 e 3, onde $\mathcal{D}_2 = 1$.

(a) Espaço bidimensional

(b) Espaço tridimensional

Fonte: Elaborada pelo autor.

Dado um reticulado de células dentro das quais o conjunto fractal encontra-se disposto, o método de *Box-counting* computa a dimensão fractal com base em uma estratégia de contagem da quantidade de pontos que incidem em cada célula. Diversos algoritmos implementam o cálculo do valor de $\mathcal{D}_2$ (BELUSSI; FALOUTSOS, 1995; SCHROEDER, 1991), sendo que o algoritmo LiBOC (*Linear Box Occupancy Counter*) (TRAINA JR. *et al.*, 2000) é um dos mais eficientes, possuindo custo linear em relação ao número de objetos N do conjunto de dados.

O algoritmo LiBOC considera um reticulado hiper-cúbico $\mathbb{L}$ de células de tamanho r em um espaço de dimensão E . Para cada célula i , o algoritmo computa a quantidade $C_{r,i}$ de pontos do fractal que estão dispostos na célula, isto é, realiza a contagem da “ocupância” da célula. A seguir, é calculado o valor $S(r) = \sum_i C_{r,i}^2$. O valor de $\mathcal{D}_2$ é dado pela derivada parcial de $\log S(r)$ com relação ao logaritmo do valor do lado r . Assim, o LiBOC consegue obter o valor da dimensão fractal de um conjunto de pontos plotando valores $S(r)$ em escala logarítmica para valores distintos de r , e calculando a inclinação da reta resultante.

O valor $S(r)$ deve ser computado para uma quantidade R de valores r distintos, de modo que o algoritmo LiBOC possa obter uma aproximação estatística adequada da reta. Para que não seja necessário ler o conjunto de dados diversas vezes para cada valor distinto de r , em Traina Jr. *et al.* (2000) é proposta uma estrutura em níveis, chamada *Árvore de Contagem*, onde cada nível da árvore contém um valor de r dado pela metade do valor do nível anterior. Uma vez que cada nível da estrutura corresponde a um valor distinto de r , sua profundidade é igual ao número de pontos r no gráfico resultante. Se o gráfico é linear para um intervalo adequado de valores de r , o conjunto de dados é um fractal e sua Dimensão Fractal de Correlação é dada pela inclinação da interpolação linear dos pontos do gráfico.

Para um valor r dado, somente as células que possuem ao menos um ponto são mantidas e a contagem da soma de sua ocupância $C_{i,r}$ é realizada. Desse modo, cada ponto é associado com a respectiva célula de seu nível, sem a necessidade de ser comparado com os anteriores.

Por exemplo, a Figura 18 (adaptada de Sousa *et al.* (2007b)) ilustra um reticulado e a

Figura 18 – Representação da Árvore de Contagem

respectiva Árvore de Contagem. Cada nível da estrutura representa um nível de refinamento do reticulado. Além disso, os nós da árvore armazenam a informação de contagem (C) e o ponteiro para o nível subsequente (P). Inicialmente, quando não há divisões do espaço, o nó raiz da Figura 18 armazena em seu valor de contagem a quantidade de 5 elementos, que é a quantidade de elementos presentes no reticulado. No primeiro nível, o reticulado é dividido em 4 partes (células), denotadas por [00], [01], [10] e [11]. Agora, é possível notar que o nó imediatamente abaixo da raiz armazena as informações destas 4 divisões, sendo que a célula [00] possui 2 elementos do fractal e a célula [11] possui os outros 3 elementos, e assim por diante, onde cada célula é novamente dividida em 4 partes. As células vazias não produzem ponteiros para os níveis inferiores, uma vez que não contribuem para a contagem.

O pseudocódigo encontrado no Algoritmo 5 apresenta o método LiBOC. O algoritmo LiBOC, contudo, tem a limitação de computar valores de dimensão fractal apenas para conjuntos de elementos finitos, sendo, portanto, inapropriado para o processamento de fluxos de dados. Na próxima Seção 4.3, é apresentada uma técnica voltada para o processamento de fluxos de dados contínuos.

4.3 Dimensão Fractal de Fluxos de Dados

Devido à sua característica infinita e evolutiva, o comportamento de um fluxo de dados pode sofrer alterações ao longo do tempo. Por exemplo, em uma rede de sensores que aferem as temperaturas mínima e máxima, é esperado que os valores coletados se alterem ao longo do ano, refletindo a sazonalidade de suas estações. Consequentemente, esta alteração nos atributos do fluxo de dados coletado reflete-se na dimensão intrínseca do conjunto.

Como discutido na Seção 4.2 anterior, a dimensão intrínseca de um conjunto de dados

Algoritmo 5 – *Linear Box Occupancy Counter* - LiBOC

```

1: função LiBOC( $\mathbb{A}$ )                                ▷ O conjunto de entrada  $\mathbb{A}$ .
Saída:  $\mathcal{D}_2$                                        ▷ A Dimensão Fractal de Correlação  $\mathcal{D}_2$ .
2:   para todo ponto do conjunto faça
3:     incremente o contador  $p_{i,r_0}$  para a célula única de lado  $r_0 = 1$ 
4:     acesse a célula apontada pela célula  $i$  no próximo nível da estrutura
5:     para  $j \leftarrow 1$  até  $R$  faça
6:        $r \leftarrow \frac{1}{2^j}$ 
7:       determine em qual célula  $i$  o ponto incide
8:       incremente o contador  $p_{i,r}$  desta célula
9:       acesse a célula apontada pela célula  $i$  no próximo nível
10:    fim para
11:    calcule a soma do quadrado das contagens da ocupação:  $S_2(r) = \sum_i p_{i,r}^2$ 
12:  fim para
13:  defina o gráfico (box-count plot) usando os valores de  $\log r$  e  $\log S_2(r)$  como abscissas e
    ordenadas, respectivamente
14:  retorne a inclinação da porção linear do gráfico como valor de  $\mathcal{D}_2$ 
15: fim função

```

finito pode ser aproximada pela Dimensão Fractal de Correlação $\mathcal{D}_2$. Quando se manipula fluxos de dados, este valor também pode ser aproximado pela $\mathcal{D}_2$. Contudo, são duas as razões que tornam esta estimativa ineficiente para este tipo de dado complexo:

- O cálculo da dimensão fractal considera que todos os dados estão disponíveis, o que não é verdade no caso de fluxos de dados, onde novas informações são recebidas continuamente.
- O reticulado definido pelas técnicas de contagem possui tamanho finito, determinado pelo hipercubo em que os dados estão imersos. Assim, uma vez que o reticulado seja definido, não é permitido que existam pontos externos à esta estrutura, ou seja, não é possível tratar novos eventos.

Visando fornecer uma solução à este impasse, o trabalho apresentado em [Sousa et al. \(2007a\)](#) propôs um algoritmo capaz de estimar a dimensão fractal de fluxos de dados multidimensionais ao longo do tempo. Esta técnica, denominada *Stream Intrinsic Dimension Meter* (*SID-meter*) é capaz de detectar mudanças em fluxos de dados multidimensionais evolutivos, baseado na informação do comportamento intrínseco fornecido pela dimensão fractal de correlação $\mathcal{D}_2$, que pode ser atualizada continuamente à requisição da aplicação que gerencia o *stream*. Para se adequar a característica evolutiva e infinita de geração de dados dos fluxos de dados, o *SID-meter* mede continuamente ao longo do tempo o valor de $\mathcal{D}_2$, com o objetivo de monitorar o comportamento evolutivo dos dados, tal que variações significantes em medidas sucessivas de $\mathcal{D}_2$ podem indicar mudanças nas características intrínsecas, bem como, na correlação entre os atributos do fluxos de dados.

Figura 19 – Representação de um hipercubo bidimensional com três níveis de estrutura e da Árvore de Contagem Continuada.

O *SID-meter* utiliza uma janela deslizante para determinar a quantidade de dados a ser utilizada para computar o valor de $\mathcal{D}_2$. Essa janela deslizante é dividida em períodos de contagem η_c e cada η_c possui um número delimitado de dados, chamados de eventos e .

Uma vez que o *SID-meter* é baseado na estratégia de *Box-counting*, seu primeiro passo é responsável por definir um hipercubo inicial que delimite os eventos e que se encontram armazenados no primeiro η_c . Os primeiros eventos do fluxo de dados são armazenados sequencialmente em memória primária até que uma quantidade η_e tenha sido recebida, possibilitando a geração de um hipercubo. Para isso, os menores e os maiores valores rL_i e rH_i de cada atributo a_i na sequência η_e são computados, sendo que o intervalo $r_0 = \max_{i=1}^E (rH_i - rL_i)$ é usado para determinar o tamanho do lado do hipercubo.

A estrutura de reticulado é criada de modo análogo ao realizado no algoritmo LiBOC. São gerados R reticulados E -dimensionais, tal que R define o número de pontos contidos no *box-count plot*. Dessa forma, a cada nível j , sendo $j \in [0, R - 1]$, as células possuem medida de lado $r_j = \frac{r_{j-1}}{2}$ e cada célula é dividida em até 2^E células no nível subsequente. A estrutura que implementa o reticulado do *SID-meter* é chamada de *Árvore de Contagem Continuada*, por permitir a contagem da ocupação ao longo do tempo.

A Figura 19 exemplifica um hipercubo criado para delimitar seis eventos de um fluxo de dados bidimensional. O lado r_0 do hipercubo delimitado é determinado pelo intervalo $[rL_x; rH_x]$, dos atributos representados no eixo x .

Cada nível j na árvore corresponde às grades de célula de lado r_j e cada nó corresponde à célula em si. Os períodos de contagem são representados por vetores $C[]$ com η_c contadores, onde existe um contador para cada período k considerado. Cada vetor $C[]$ pode ser percebido como uma lista circular, onde um índice k_c define qual contador acumula a contagem de ocupação

Figura 20 – Fluxo de dados de 3 dimensões com os eventos $e_i = (a_1, a_2, a_3)$ recebidos em períodos de contagem de uma janela móvel.

Fonte: Elaborada pelo autor.

(quantidade de eventos) de cada célula do reticulado, considerando os eventos recebidos no período de contagem do índice k_c . Dessa forma, quando um determinado período expira, o próximo contador da lista passa a ser utilizado, iniciando a contagem de eventos do período seguinte. Assim, quando após η_c períodos, os eventos anteriores são descartados e o próximo contador é disponibilizado para computar a ocupação de novos eventos, de acordo com a Figura 20a-d (SOUSA *et al.*, 2007b).

Os primeiros η_e eventos que foram usados para definir o hipercubo inicial são considerados no primeiro período de contagem. Após todos os contadores serem atualizados, significa que estes eventos foram processados e, portanto, podem ser descartados, sendo que a memória por eles ocupada é então liberada.

Após o processamento de cada período, a Dimensão Fractal de Correlação $\mathcal{D}_2$ (Equação 4.2) é calculada para determinar a dimensão intrínseca dos dados da janela móvel. A contagem da ocupação C_{i,r_j} para cada nó i de cada nível j é dada a partir da soma dos valores acumulados nos contadores $C[\]$, de acordo com a Equação 4.3.

$$C_{i,r_j} = \sum_{k=0}^{k < \eta_c} C_i[k] \quad (4.3)$$

Assim, a soma dos quadrados das contagens de ocupação realizadas na Equação 4.2 é

expressa como $S_2(r_j) = \sum_{i=0}^{i < 2^E} C_{i,r_j}^2$. Já o *box-counting plot* é criado percorrendo-se toda a Árvore de Contagem Continuada e plotando $\langle \log(S_2(r_j)), \log(r_j) \rangle$ para cada valor r_j da árvore. Então, a inclinação da linha que melhor se ajustar a curva fornece uma estimativa de $\mathcal{D}$ para os eventos dentro da janela deslizando.

Dessa forma, a estratégia de janela deslizando dividida em períodos de contagem permite com que o valor da dimensão intrínseca se altere ao longo do tempo e reflita as alterações de comportamento causadas no fluxo de dados, quando novos eventos são percebidos e os anteriores descartados, de acordo com a evolução dos dados.

Outro atrativo do SID-Meter é sua complexidade computacional. Essa técnica realiza o cálculo da dimensão intrínseca dos eventos contidos numa janela móvel, em tempo computacional proporcional a $\mathcal{O}(e \cdot \eta_c \cdot E \cdot R)$, onde e é a quantidade de eventos, n_c o número de períodos de contagem, E é o espaço E -dimensional e R é a quantidade de pontos no gráfico *box-counting plot*.

Além disso, o *SID-meter* mantém a Árvore de Contagem Continuada em memória primária de modo a obter um melhor desempenho no processo de cálculo da dimensão intrínseca. Uma vez que os nós da árvore são criados apenas quando uma célula correspondente do reticulado está ocupada por ao menos um evento, a quantidade de nós em cada nível é superiormente limitada à quantidade de eventos contidos na janela deslizando, ou seja, $\eta_e \cdot \eta_c$. Desse modo a complexidade computacional espacial para o armazenamento da árvore é proporcional a $\mathcal{O}(\eta_e \cdot \eta_c \cdot R)$.

Devido às características apresentadas, o cálculo da dimensão fractal foi utilizado como uma componente importante nos métodos desenvolvidos nesse doutorado, com o objetivo de melhor identificar a correlação entre as dimensões dos fluxos de dados, buscando capturar o comportamento de fluxos de dados multidimensionais ao longo do tempo. Esse comportamento é utilizado para mensurar a similaridade entre os fluxos de dados com o intuito de melhorar as formações de agrupamento, como poderá ser observado nos resultados apresentados no Capítulo 6.

4.4 Considerações finais

Este capítulo apresentou os aspectos mais importantes da Teoria de Fractais. Foram descritos os conceitos de fractais e de dimensão fractal, que permite determinar a dimensão intrínseca de um conjunto e encontrar correlações existentes entre os dados.

A dimensão intrínseca de conjuntos finitos pode ser obtida através do cômputo da Dimensão Fractal de Correlação $\mathcal{D}_2$. Os algoritmos responsáveis por este processamento frequentemente baseiam-se na técnica de *Box-counting*, sendo que o algoritmo LiBOC é um dos mais clássicos encontrados na literatura.

Por sua vez, devido às suas características peculiares, a dimensão intrínseca de um fluxo de dados não pode ser obtida de modo semelhante à dos conjuntos finitos. Para realizar esse cálculo, o assim chamado método *SID-meter* permite acompanhar a evolução deste tipo de dado através de janelas deslizantes e acompanhar as suas alterações ao longo do tempo. O próximo Capítulo 5 descreve as propostas desta tese, os métodos eFCDS e eFCC, que são embasados na Teoria de Fractais.

AGRUPAMENTO DE FLUXOS DE DADOS E DETECÇÃO DE *OUTLIERS* BASEADO EM FRACTAIS

Conforme apresentado no capítulo 3, na literatura foram propostos diversos métodos para extrair conhecimento de fluxo de dados por meio da construção de agrupamentos. Esses métodos utilizam as mais diversas estratégias de agrupamento, por exemplo, por particionamento, construindo uma hierarquia ou utilizando uma estrutura de grade.

Em geral, esses métodos agrupam dados ponto a ponto, provenientes de um único fluxo de dados. Conforme apresentado na seção 3.1.4, são poucos os métodos que realizam o agrupamento de fluxos de dados. Dentre eles, apenas o algoritmo POD-Clus (CHAOVALIT, 2012) trata fluxos de dados multidimensionais, mas sem considerar a correlação entre dimensões e apresentando uma solução pouco eficaz para identificar fluxos de dados *outliers*. Além disso, a maioria desses métodos apresenta abordagens baseadas em medidas estatísticas descritivas dos dados, tais como: medidas de tendência central, medidas de dispersão ou medidas que mensuram a distribuição dos dados. Ao utilizar tais tipos de medidas os métodos não conseguem capturar o comportamento dinâmico presente nos fluxos de dados, conforme apresentou Cormode *et al.* (2012) em seus estudos.

Diante desse cenário, o presente trabalho de doutoramento teve por meta desenvolver métodos de agrupamento de fluxos de dados que lidassem com a multidimensionalidade e as correlações presentes nos dados, que representassem o comportamento dinâmico dos dados de modo incremental e que trate fluxos que apresentarem algum desvio de comportamento dentro do período analisado.

Sendo assim, dois novos métodos foram desenvolvidos. O primeiro, chamado eFCDS (evolving Fractal Clustering of *Data Streams*), que realiza o agrupamento de fluxos de dados multidimensionais calculando a dimensão fractal dos fluxos de dados para agrupar os de compor-

tamento mais similar. O eFCDS ainda faz o monitoramento dos agrupamentos para acompanhar a evolução dos dados. O segundo método, chamado eFCC (evolving Fractal Clusters Construction), realiza a construção dos agrupamentos calculando a dimensão fractal do grupo em si, de modo que um fluxo de dados só poderá ser inserido, no grupo em análise, se a dimensão fractal do grupo não sofrer alterações significativas após a inserção. O eFCC também realiza a atualização dos agrupamentos sempre que identificar que um fluxo de dados deve ser remanejado. Além disso, tanto o eFCDS quanto o eFCC monitoram o aparecimento de *outlier* e dão o devido tratamento a eles.

Embora os dois métodos propostos tenham como base a dimensão fractal, esse conceito é utilizado em abordagens distintas, visando propósitos diferentes para a tarefa de agrupamento de fluxos de dados. O eFCDS utiliza a dimensão fractal como uma medida sumarizada dos dados de um fluxo de dados multidimensional. Essa medida é calculada individualmente para cada fluxo de dados, de modo que fluxos de dados com valores próximos de dimensão fractal são inseridos num mesmo grupo. Em termos práticos, isso significa que as correlações entre dimensões de um determinado fluxo de dados são similares às correlações entre dimensões dos demais fluxos de dados alocados num mesmo grupo. Como ilustração de um cenário de aplicação, considere o exemplo de sensores agrometeorológicos mencionado no Capítulo 1, em que cada sensor coleta temperatura mínima e temperatura máxima: o eFCDS colocaria num mesmo grupo sensores em que temperatura mínima e máxima estivessem linearmente correlacionadas, e num outro grupo sensores em que temperatura mínima e máxima fossem completamente independentes, por exemplo. Vale destacar que o eFCDS considera a correlação entre as dimensões do fluxo de dados e não os valores individuais de cada dimensão em cada ponto de dado. No exemplo, sensores provenientes de regiões muito frias e sensores de regiões muito quentes poderiam ser atribuídos a um mesmo grupo se temperatura mínima e máxima fossem linearmente correlacionadas em ambos os casos. Nesse caso, a análise do especialista a cerca dos resultados da tarefa de agrupamento teria como foco a avaliação de comportamento das correlações entre variáveis climáticas em diferentes regiões, identificando comportamentos similares e as modificações dessas correlações ao longo do tempo, independente dos valores propriamente ditos coletados para essas variáveis.

Numa segunda abordagem, o eFCC trata, além das correlações entre dimensões de cada fluxo de dados, a distribuição dos dados no grupo, avaliando o comportamento das correlações entre dimensões (i.e., a dimensão fractal) considerando todos os pontos de dados de todos os fluxos de dados do grupo, de maneira integrada. Logo, fluxos de dados atribuídos a um mesmo grupo devem possuir também distribuições e intervalos de valores similares no espaço multidimensional em que estão definidos. Em outras palavras, a inserção ou remoção de cada um dos fluxos de dados do grupo não causaria alterações significativas no valor da dimensão fractal do grupo como um todo. No cenário de aplicação utilizado como exemplo, dois sensores com temperatura máxima e mínima linearmente correlacionadas só seriam alocados no mesmo grupo se a distribuição e o intervalo de valores dessas variáveis climáticas fossem similares. Nesse caso,

a análise do especialista teria como foco, por exemplo, identificar regiões com características climáticas similares ao longo do tempo.

Esse Capítulo descreve na Seção 5.1 o funcionamento do eFCDS, na Seção 5.2 é apresentado o método eFCC em detalhes, e na Seção 5.3 são apresentadas as considerações finais do Capítulo.

5.1 eFCDS

O método eFCDS (*evolving Fractal-Based Clustering of Data Streams*) surgiu da necessidade de preencher as seguintes lacunas deixadas pelos métodos de agrupamento de fluxos de dados apresentados na Seção 3.1.4: não tratam as correlações entre as dimensões dos fluxos de dados multidimensionais para construir os agrupamentos, serem muito sensíveis a *outliers* por utilizarem sumarização dos dados, tenderem a encontrar grupos de forma arredondada ou elipsoidal (*spherical shapes*) e tempo de processamento elevado.

Desse modo, o eFCDS foi projetado para criar grupos disjuntos $\mathbb{C}$, cujos membros de um grupo não possam estar inseridos em outro grupo conforme definição 3 apresentada na Seção 3.1.2, amorfos (sem forma definida) de um conjunto de fluxos de dados multidimensionais $\mathbb{S}^D$, conforme definição 5 apresentada na Seção 3.1.4, realizando o cálculo de dimensão fractal dos fluxos de dados. O eFCDS lida também com os problemas de: 1) redução da sobreposição entre agrupamentos com junção ou divisão dos grupos; e 2) detecção de *outliers* buscando por potenciais grupos de *outliers* (como exemplo, grupos com um único fluxo de dados) que não possam ser integrados com seus grupos próximos (não *outlier*).

O eFCDS foca em particionar o conjunto $\mathbb{S}^D$ em uma coleção $\mathbb{C}$ de grupos disjuntos, de acordo com a definição 5 apresentada na Seção 3.1.4, utilizando como base a análise da dimensão fractal de cada fluxo de dados multidimensional DS_i , introduzido em 2 da Seção 2.1. Cada grupo C é composto de fluxos de dados considerados similares entre si se eles não excedem um parâmetro de desvio padrão máximo definido pelo usuário. O eFCDS possibilita agrupar fluxos de dados com comportamento similar em um intervalo de tempo T_i , considerando as correlações entre as dimensões mensuradas por meio da dimensão fractal $\mathcal{D}_2$ (Equação 4.2). O método proposto também segue a evolução dos fluxos de dados em que grupos podem desaparecer ou serem criados ao longo do tempo.

A ideia geral do eFCDS é ilustrada na Figura 21. As fontes de fluxos de dados (como sensores) geram continuamente novos dados e os enviam para o eFCDS que retorna os agrupamentos evolutivos gerados. Cabe destacar que, para cada intervalo de tempo T_i , os fluxos de dados são agrupados baseados na dimensão fractal dos dados disponíveis. Com a chegada de novos dados, os grupos são criados ou rearranjados, ou distribuídos, para garantir a similaridade entre os fluxos de dados e a evolução das correlações entre as dimensões ao longo do tempo. Por exemplo, na Figura 21, do intervalo T_1 para T_2 um grupo é desfeito e no intervalo T_2 para T_3 dois

Figura 21 – Exemplo ilustrado de agrupamento de fluxos de dados, no qual cada SX_{FD} representa um sensor que é agrupado usando a medida de dimensão fractal $\mathcal{D}_2$ no tempo T_i .

novos grupos são criados.

Os principais componentes do eFCDS são apresentados na Figura 22. O processo inicia com um determinado número de fontes de fluxo de dados escolhidas para análise. Conforme as fontes produzem dados, esses podem ser diretamente encaminhados ao eFCDS para serem processados.

O componente de entrada do eFCDS é um módulo de janela deslizante de tamanho fixo. Este módulo recebe os fluxos de dados e restringe as informações pelo janelamento pré-definido. O janelamento definido especifica a quantia de informação guardada para realizar o cálculo de dimensão fractal. A janela é dividida em períodos de contagem (η_c) e cada período tem uma quantidade e de eventos de modo que cada evento representa um ponto de dado multidimensional x_i^d . Por isso, $l = \eta_c \times e$ define o tamanho da janela e e também representa o deslocamento da janela. A janela tem tamanho padrão l , o qual geralmente considera conhecimento de especialista do domínio ou a sazonalidade dos dados. A Figura 23 contém a janela deslizante w_i de tamanho $l = 12$, resultado de quatro períodos de contagem ($\eta_c = 4$), tal que cada período tem três eventos ($e = 3$).

Assim que a quantidade de eventos e preenche a janela, esses dados podem ser processados com uma leitura única, como exemplo, o instante T1 na Figura 23. Neste caso, a janela

Figura 22 – Visão geral do método eFCDS para agrupar fluxos de dados.

Figura 23 – A janela deslizante (W) de um sensor, períodos de contagem $\eta_c = 4$, eventos $e = 3$. No instante T_1 , a janela inicia no período 1 e termina no período 4; depois deslocando a janela para T_2 , (W) inicia no período 2 e termina no período 5.

Fonte: Elaborada pelo autor.

deslizante é deslocada em e unidades para processar novos dados de entrada, liberando e desconsiderando os dados mais antigos, que ficam fora do escopo da janela após o deslocamento. No exemplo da Figura 23, no instante T_2 a janela é deslocada para incorporar dados do período de contagem 5, mais recente, desconsiderando dados do período 1, mais antigo. Ao ser preenchida, os dados da janela são enviados para o módulo responsável por executar a análise de dimensão fractal.

O processo de cálculo da dimensão fractal considera uma abordagem análoga à aplicada pelo SID-Meter, apresentado na Seção 4.3, para calcular $\mathcal{D}_2$ de modo incremental. Ao contrário dos métodos de agrupamento de fluxos de dados propostos na literatura, que tratam as dimensões de maneira independente, essa abordagem permite analisar e sumarizar fluxos de dados multidimensionais considerando as correlações existentes entre as dimensões. Assim, esta etapa do processamento sumariza uma parte do fluxo de dados multivalorado de tamanho l em um valor de dimensão fractal DF , que representa o comportamento das correlações no período de tempo em questão. Essa medida de dimensão fractal, chamada neste Capítulo de ponto de dimensão fractal, é gerada para cada fluxo de dados de entrada, a cada instante T_i definido pelo deslocamento da janela temporal, gerando portanto um conjunto de pontos de dimensão fractal a ser enviado ao módulo de mineração de dados.

A Figura 24 ilustra, para o sensor de Sete Lagoas (introduzido na Seção 1.1), a redução no volume de dados que é transferido para a etapa de agrupamento propriamente dita: a cada deslocamento da janela (t_i), 1095 itens de dados tridimensionais são sumarizados em um valor $DF_2(\text{SeteLagoas})$.

O módulo de mineração é o principal componente do eFCDS e objetiva agrupar os pontos de dimensão fractal de acordo com sua similaridade. Para isso, é utilizada a distância *Manhattan* (L1) e o parâmetro σ_{MAX} , inserido pelo usuário para determinar o desvio padrão máximo permitido para cada grupo. Esse módulo principal efetua a construção dos agrupamentos com base no pseudo-código descrito no algoritmo 6.

Quando o algoritmo 6 inicia o seu processamento, não há nenhum grupo formado, ou

Figura 24 – Exemplo de redução de dados transferido para o módulo de mineração pelo módulo de análise de dimensão fractal. Em cada passo da janela deslizante (t_i) é transferida apenas uma dimensão fractal para o módulo de mineração.

Fonte: Elaborada pelo autor.

Algoritmo 6 – Agrupamento evolutivo baseado em fractal de *Data Streams* (eFCDS)

- 1: **função** eFCDS($X, \mathbb{C}, \sigma_{MAX}$) $\triangleright$ Conjunto X contendo pares (DS_i, DF) representando o fluxo de dados e sua dimensão fractal, respectivamente; Partição de agrupamentos $\mathbb{C}$ tal que na primeira iteração $\mathbb{C} = \{\}$; Desvio padrão máximo σ_{MAX} .

Saída: Partição $\mathbb{C}$ de agrupamentos com fluxos de dados similares.

- ```

2: para todo $x \in X$ faça
3: se (DS_i, x_i) pertence a um grupo de $C_j \in \mathbb{C}$ então
4: se UPDATECLUSTER(C_j, x, σ_{MAX}) = falso então
5: se FINDCLUSTER(x, σ_{MAX}) = falso então
6: $C_{new} \leftarrow$ CREATENEWCLUSTER(x)
7: REARRANGE TO CLOSEST(C_j, C_{new})
8: fim se
9: fim se
10: senão se FINDCLUSTER(x, σ_{MAX}) = falso então
11: $C_{new} \leftarrow$ CREATENEWCLUSTER(x)
12: fim se
13: fim para
14: OUTLIER($\mathbb{C}$)
15: OVERLAP($\mathbb{C}$)
16: retorna $\mathbb{C}$
17: fim função

```
- 

seja,  $\mathbb{C} = \emptyset$ . Logo, o algoritmo 6 começa à iterar sobre o conjunto  $X$ , composto de pontos de dimensão fractal ( $DF$ ), tal que cada ponto  $x$  é rotulado com seu respectivo fluxos de dados  $DS_i$ . Inicialmente é verificado se o fluxos de dados  $DS_i$  pertence ou não a algum agrupamento  $C_i$  (Alg.: 6, linha 3), nesse ponto duas situações podem ocorrer: 1) o fluxo de dados ainda não foi agrupado e, 2) o fluxo de dados já foi agrupado em alguma iteração prévia.



**Situação 1)** supondo que  $DS_i$  ainda não foi agrupado, então é necessário procurar por um grupo  $C \in \mathbb{C}$  para inserir  $DS_i$ . Essa tarefa é realizada pela função `findCluster` (Alg.: 6, linha 10), que irá procurar pelo grupo  $C_j$  com a menor diferença entre o centroide  $\Delta$  de  $(C_j)$  e o ponto de dimensão fractal  $x$  em análise. Para que isso ocorra, considere que  $C \in \mathbb{C}$  e é composto pelos  $N$  pares  $(DS, x)$  de todos os fluxos de dados adicionados a  $C$  de tal modo que  $C = \{(DS_1, x_1), (DS_2, x_2), \dots, (DS_n, x_n) \mid 1 \leq n \leq N\}$ . Então, utiliza-se a Equação 5.1 para calcular a diferença da média das dimensões fractais em  $C$  para o valor da dimensão fractal de  $DS_i$  em análise. Uma vez encontrado o grupo  $C$  que minimiza a Equação 5.1 em relação ao elemento  $x$ , então,  $x$  é associado a  $C$  se, e somente se, a condição expressa na Equação 5.2 for mantida. Caso contrário, um novo grupo  $C_{new}$  contendo o elemento  $x$  é criado (Alg.: 6, linha 11) e  $C_{new}$  adicionado no conjunto  $\mathbb{C}$  de grupos.

$$\Delta(C, x) = \left( \frac{1}{N} \sum_{n=1}^N x_n \cdot df \right) - x \cdot df \quad (5.1)$$

$$\sqrt{\frac{1}{n+1} \left( \left( x \cdot DF - \left( \frac{1}{n} \sum_{p=1}^n c_p \cdot DF \right) \right)^2 + \sum_{k=1}^n \left( c_k \cdot DF - \left( \frac{1}{n} \sum_{p=1}^n c_p \cdot DF \right) \right)^2 \right)} \leq \sigma_{MAX} \quad (5.2)$$

**Situação 2)** considerando agora que os fluxos de dados  $DS_i$  já é membro de algum grupo  $C_j$ , então, é indispensável checar se  $DS_i$  permanecerá no grupo ao qual ele pertence ou não. Com o intuito de realizar tal verificação (Alg.: 6, linha 4), o algoritmo re-executa o cálculo da Equação 5.2 considerando o novo valor de dimensão fractal  $x \cdot DF$ , calculado no módulo de análise de dimensão fractal para o instante de tempo atual. Se essa equação for satisfeita, então a estatística (função  $\Delta$ ) do grupo  $C_j$  é atualizada. Caso o valor mensurado na parte esquerda da Equação 5.2 exceda o valor definido pelo usuário para o desvio padrão máximo  $\sigma_{MAX}$ , o algoritmo tentará realocar o par de valores  $(DS_i, x_i)$  em um outro grupo existente  $[C_k \in \mathbb{C} \mid \forall k \neq j]$ , que minimize o valor computado na Equação 5.1 (Alg.: 6, linha 5) e não extrapole a condição definida na Equação 5.2. Caso não exista nenhum grupo que possa acomodar  $(DS_i, x_i)$  sem violar as duas condições, então um novo grupo  $C_{new}$  é criado para incluir  $(DS_i, x_i)$  (Alg.: 6, linha 6). Ao criar o novo grupo para acomodar  $(DS_i, x)$  os elementos remanescentes no grupo  $C_j$  serão checados para verificar se eles permanecem onde estão ou se seria melhor incluí-los no novo grupo  $C_{new}$  sem ferir as condições impostas pelas equações 5.1 e 5.2 (linha 7). Criando esse novo grupo e realocando adequadamente os fluxos de dados de  $C_j$  para  $C_{new}$ , o eFCDS esta promovendo a divisão de  $C_j$ .

Após processar todo  $X$  e alocar cada fluxo de dados em seu respectivo grupo, o eFCDS busca por grupos cuja quantidade de  $(DS_i, x_i)$  inseridos seja uma pequena fração  $f$  de  $X$ , visando identificar *outliers*. Essa fração pode ser definida pelo usuário, que se não indicá-la é atribuído o valor empírico de  $(\lceil |X|/100 \rceil) > 0 ? (\lceil |X|/100 \rceil) : 1$ . Portanto, o procedimento de OUTLIER

(Alg.: 6, linha 14) irá percorrer  $\mathbb{C}$  para identificar grupos com fração menor ou igual a  $f$ , aplicando a esses grupos a função FINDCLUSTER a todos os seus membros.

Figura 25 – (a) Sobreposição entre os grupos; (b) Primeiro tenta realizar a união dos grupos sem extrapolar o máximo desvio padrão  $\sigma_{MAX}$ ; (c) Se falhar na união, então um novo grupo é criado e é testado se os fluxos de dados permanecerão no seu grupo original ou migrarão para o novo grupo.

Fonte: Elaborada pelo autor.

O último passo do algoritmo 6 é verificar se existem grupos que estão em sobreposição (Alg.: 6, linha 15). Grupos com sobreposição significativa são unidos de acordo com o algoritmo MERGECLUSTER (Alg.: 7). Inicialmente o algoritmo MERGECLUSTER determina se existe ou não grupos que podem ser unidos por meio do cálculo do diâmetro de cada grupo, isto é, a diferença dos dois fluxos de dados mais distantes do grupo. O diâmetro de cada grupo é comparado ao diâmetro dos outros grupos para identificar se existe sobreposição entre eles, como apresentado na Fig. 25(a). Se a sobreposição é detectada, os grupos que estiverem em justaposição são catalogados (Alg.: 7:linha 2) e unidos se a sua união não exceder o limite definido pela Equação 5.2 (Alg.: 7, linha 4), conforme apresentado na Fig. 25(b). Nesse caso, um grupo deixa de existir. Caso contrário, são encontradas as bordas dos grupos  $i$  e  $j$  (Alg.: 7, linha 5) encontrando o  $DS_i$  que está mais próximo do centroide de  $C_j$  e o  $DS_j$  que está mais próximo do centroide de  $C_i$ . Então,  $DS_i$  e  $DS_j$  serão adicionados a um novo grupo  $C_{ij}$  (Alg. 7, linha 6) e todos os fluxos de dados remanescentes em  $C_i$  e  $C_j$  serão analisados para apurar se eles permanecerão em seus respectivos grupos ou se eles serão inseridos no novo grupo  $C_{ij}$  (Alg.: 7, linha 7), de acordo com o apresentado na Fig. 25(c). Conforme pode ser observado, o algoritmo *MergeCluster* não promove apenas a união de grupos, mas também a criação de um novo agrupamento quando for relevante.

Assim, ao final de uma iteração sobre o conjunto  $X$ , a partição  $\mathbb{C}$  é composta por grupos disjuntos contento fluxos de dados similares entre si, em relação à dimensão fractal (Alg.: 6, linha 16). A medida que as fontes de dados geram, continuamente, novas medidas, a janela deslizante  $W$  se desloca 1 período de contagem para acomodar  $e$  novos eventos. Assim que os novos eventos são preenchidos em  $W$ , uma nova dimensão fractal  $\mathcal{D}_2$  é gerada e o módulo de mineração é reexecutado. Desse modo, por meio das operações de divisão, união e criação de novos grupos, o eFCDS oferece suporte incremental à evolução dos grupos ao longo do tempo, refletindo as mudanças de comportamento dos dados ao longo do tempo.

**Algoritmo 7** – Função para unir grupos em sobreposição

---

```

1: função MERGECLUSTER($\mathbb{C}$, σ_{MAX}) $\triangleright$ Partição $\mathbb{C}$ de grupos, Desvio Padrão Máximo σ_{MAX} .
Saída: Partição P de grupos com fluxos de dados similares.
2: $ov[] \leftarrow \text{FINDOVERLAP}(\mathbb{C})$
3: para todo $pair(i, j) \in ov$ faça
4: se $\text{MERGE}(i, j) > \sigma_{MAX}$ então
5: $\text{FINDMARGIN}(i, j)$
6: $C_{new} \leftarrow \text{CREATENEWCLUSTER}(ij)$
7: $\text{CHECKREALLOC}(C_i, C_j, C_{ij})$
8: fim se
9: fim para
10: retorna P
11: fim função

```

---

Todo o processo de cálculo da dimensão fractal  $\mathcal{D}_2$ , utilizando uma janela deslizante, é a maior complexidade do método de  $O(l \times E \times R \times |\mathbb{C}|)$  tal que,  $e$  é o número de eventos num período de contagem  $\eta_c$ ,  $E$  a dimensão de imersão,  $R$  a quantidade de células ocupadas com pelo menos um ponto de dados no hiperreticulado e  $|\mathbb{C}|$  a quantidade de grupos disjuntos.

## 5.2 eFCC

O eFCC (*evolving Fractal Clusters Construction*) foi desenvolvido, assim como o eFCDS, para realizar o agrupamento de fluxos de dados multidimensionais e que possuam comportamento semelhante ao longo do tempo. Entretanto numa abordagem diferente, o eFCC agrupa os fluxos de dados que apresentam distribuição similar dos dados no mesmo período de tempo. Para isso é calculada a dimensão fractal de correlação  $\mathcal{D}_2$  para o grupo como um todo. O eFCC foi inspirado no trabalho de [Cormode et al. \(2012\)](#), que utiliza como critério de agrupamento o valor da dimensão fractal dos grupos gerados. Todavia, o método proposto em ([CORMODE et al., 2012](#)), é voltado a agrupamento de dados, ponto a ponto, e requer mais de uma leitura nos dados para realizar a divisão de um grupo formado, sendo necessário, inclusive, recuperar os dados no disco rígido. O eFCC, por outro lado, realiza o agrupamento de fluxos de dados com apenas uma leitura dos dados para qualquer operação (união, divisão ou construção) nos grupos gerados.

Desse modo, o objetivo principal do eFCC é criar grupos amorfos disjuntos de  $\mathbb{S}^D$ , com relação à análise  $\mathcal{D}_2$  de  $C_j$  -  $DF_2(C_j)$  - e da dimensão fractal de cada  $DS_j$ . A definição de conjunto disjunto foi apresentada na definição 3. E a distribuição dos dados é considerada semelhante se, ao adicionar um novo  $DS^{new}$  a  $C_j$  a condição expressa pela Equação 5.3, for satisfeita.

$$|DF_2^{new}(C_j) - DF_2(C_j)| \leq (\tau * DF_2(C_j)) \quad (5.3)$$

Onde  $\tau$ , um parâmetro fornecido pelo usuário, indica o valor de expansão ou contração de  $C_i$ , sendo uma porcentagem da dimensão intrínseca  $\mathcal{D}_2$  no intervalo  $[0, 1]$  e, a dimensão fractal do agrupamento é representada pelo valor  $DF_2$ .

Figura 26 – Processo para verificar se um fluxo de dados pode ser incluído em um grupo baseando-se na expansão do mesmo.

Fonte: Elaborada pelo autor.

A Figura 26 apresenta esse conceito de expansão e contração ao qual os agrupamentos são submetidos para acomodar novos fluxos de dados. Na primeira etapa, Figura 26a, os fluxos de dados  $DS_1$  e  $DS_2$  se encontram alocados no grupo  $C_1$  (com valor de dimensão fractal igual a  $DF_2(C_1)$ ) e  $DS_3$  no grupo  $C_2$ . Ao chegar um novo fluxo de dados ( $DS_4$ ), Figura 26b, primeiramente é verificado se ele pode ser inserido no grupo  $C_1$  por possuir dimensão fractal semelhante a  $DF_2(C_1)$ . Todavia, para acomodar  $DS_4$  a expansão verificada de  $DF_2(C_1)$  extrapolaria o parâmetro  $\tau$ . Sendo assim, o eFCC tentará inserir  $DS_4$  no grupo  $C_2$ , Figura 26c, e como o aumento de  $DF_2(C_2)$  é menor que o definido na Equação 5.3,  $DS_4$  será inserido em  $C_2$ , efetivando a alteração do valor da dimensão fractal  $DF_2(C_2)$ , Figura 26d, até que novo fluxo de dados seja inserido em  $C_2$ . Caso  $C_2$  não pudesse agrupar  $DS_4$  e não havendo mais grupos para serem testados, o fluxo de dados  $DS_4$  formaria sozinho um novo grupo à espera de novos fluxos de dados que pudessem se juntar a ele. O processo de contração (diminuição da dimensão fractal) de  $DF_2(C_i)$  é análogo ao processo de expansão, ou seja, se ao tentar inserir um fluxo de dados em um agrupamento  $C_i$  a diminuição do valor  $DF_2(C_i)$  seja maior que o permitido pela Equação 5.3, então  $C_i$  não poderá acomodar o fluxo de dados, tornando-se necessário realizar o teste em um outro grupo.

Os principais componentes do eFCC são apresentados na Figura 27. Os principais parâmetros utilizados pelo eFCC são:

- **Número de períodos de contagem  $\eta_c$** : esse parâmetro determinará o deslocamento da janela deslizante a cada iteração.
- **Número de pontos de dados por  $\eta_c$** : esse parâmetro indica quantos pontos serão recebidos e quantos pontos de dados serão descartados a cada deslocamento. A quantidade de períodos de contagem  $\eta_c$  vezes o número de ponto de dados determina o tamanho total da janela deslizante. Poucos pontos de dados por período de contagem podem levar a detecção de falsos *outliers*. Todavia, ao utilizar muitos pontos de dados por período de contagem podem mascarar o comportamento dinâmico dos dados.
- **Desvio Padrão Máximo  $\sigma_{max}$** : esse parâmetro determina quão próximos ou distantes os fluxos de dados podem estar em um grupo. Valores muito pequenos, na ordem de centésimos pode levar a construção de um grupo para cada fluxo de dados. Já valores muito grandes, próximos a dimensão de imersão  $E$  do conjunto de fluxos de dados resultará no agrupamento dos fluxos de dados em um único grupo.
- **Expansão ou contração permitida para grupos  $\tau$** : Esse parâmetro determinará quanto de alteração um grupo pode sofrer na distribuição dos seus dados. Sendo uma porcentagem da dimensão de imersão  $E$  do conjunto de fluxos de dados. Novamente valores muito pequenos pode levar a construção de um grupo para cada fluxo de dados e, valores muito grandes resultará no agrupamento dos fluxos de dados em um único grupo.

O processo inicia do mesmo modo que o método eFCDS, onde um determinado número de fontes de fluxos de dados encaminha seus dados aos módulos de Janela Deslizante e de Análise

Figura 27 – Visão geral do eFCC para agrupar fluxos de dados.

Fonte: Elaborada pelo autor.

de Dimensão Fractal, conforme pode ser observado na Figura 27. Esse módulos aplicam o processamento similar ao eFCDS e por esse motivo não serão repetidos aqui.

Após o processamento dos módulos iniciais, o valor de dimensão fractal ( $DF(DS_i)$ ) e a janela deslizante  $w_i$  de cada um dos fluxos de dados são transferidos para o módulo de mineração para realizar a construção dos agrupamentos com base na semelhança e na distribuição dos dados de cada  $DS_i$ . O módulo de mineração é o principal componente do eFCC e objetiva agrupar os fluxos de dados utilizando dois critérios: 1) os pontos de dimensão fractal  $FD(DS_i)$  serão analisados de acordo com sua similaridade baseada na distancia *Manhattan* (L1) e o desvio padrão máximo  $\sigma_{MAX}$  e; 2) Utilizando o critério de cálculo  $\mathcal{D}_2$  dos grupos para verificar a expansão ou contração dos agrupamentos ao tentar adicionar um novo fluxo de dados, respeitando a determinação imposta pela Equação 5.3. Entretanto, antes de apresentar o algoritmo eFCC (Alg.: 10) que é o módulo principal da etapa de mineração, serão apresentados os algoritmos *Busca e Inserção em Grupos - BIG* e *UpdateeFCC*, pois esses algoritmos são utilizados pelo eFCC para construção dos agrupamentos.

O algoritmo *Busca e Insere em um Grupo (BIG)*(Alg.:8) procura em toda a partição  $\mathbb{C}$  para encontrar um grupo que possa acomodar  $DS_i$  (Alg.:8 linha: 3). Para isso, primeiro é verificado quais grupos possuem a dimensão fractal mais próxima da dimensão fractal de  $DS_i$  sem extrapolar o máximo desvio padrão ( $\sigma_{MAX}$ ) (Alg.:8 linha: 4). Para os grupos que atenderem esse requisito, verifica-se se podem receber o fluxo de dados sem exceder o critério de expansão ou contração (Alg.:8 linha:5), como apresentando, em detalhes, na Figura 26. Todos os grupos que atenderem os requisitos anteriores serão adicionados ao vetor *candidate[]* juntamente com a nova dimensão fractal calculada e a diferença entre a dimensão antiga e a nova (Alg.: 10 linha 6). Após testados todos os grupos de  $\mathbb{C}$ , é verificado em *candidate[]* o grupo com menor expansão ou contração e, caso exista, o  $DS_i$  é adicionado nesse grupo, a respectiva dimensão fractal é alterada para o novo valor (Alg.: 10 linha: 11) e o algoritmo *BIG* é encerrado com sucesso. Caso o vetor *candidate[]* esteja vazio, ou seja, nenhum grupo atendeu aos critérios de similaridade, contração e expansão, o algoritmo é encerrado sem sucesso (Alg.: 10 linha: 14).

Quando um fluxo de dados já estiver inserido em um grupo, resultado do processamento dos dados de um período anterior, é necessário avaliar se o mesmo poderá permanecer no grupo, ou se ocorreram mudanças no comportamento do fluxo de dados, no período atual analisado, que levaram a alteração da dimensão fractal e, consequentemente, da distribuição dos dados. Essa checagem é efetuada pelo algoritmo *UpdateeFCC* (Alg.:9). Primeiro é verificado se a nova dimensão fractal do fluxo de dados é similar à dimensão fractal do grupo atual de  $DS_i$ , o grupo  $C_j$ , e se não extrapola  $\sigma_{MAX}$  (Alg.: 9, linha: 2). Se o primeiro teste for concluído com sucesso, então é verificado quanto  $C_j$  irá expandir ou contrair com a inserção dos novos dados de  $DS_i$  (Alg.: 9, linha: 3). Se esse movimento não extrapolar o critério estabelecido na Figura 26, então o  $C_j$  é atualizado com os novos dados de  $DS_i$  e é atualizada a dimensão fractal de  $C_j$  (Alg.: 9, linha: 4), sendo o algoritmo *UpdateCluster*, encerrado com sucesso. Caso alguma das verificações falhe o

**Algoritmo 8 – Busca e Inserção de fluxo de dados em um Grupo - BIG**


---

```

1: função BIG($(DS_i, x), \mathbb{C}, \sigma_{MAX}, w_i, \tau$) ▷ Par (fluxo de dados, dimensão fractal
 DF_2); Partição de agrupamentos $\mathbb{C}$; Des-
 vio padrão máximo σ_{MAX} ; w_i conjunto de
 dados do fluxo de dados, τ parâmetro de
 expansão ou contração dos grupos.

2: $candidate \leftarrow \{, \}$
3: para todo $c \in \mathbb{C}$ faça
4: se $|x.DF_2 - c.DF_2| \leq \sigma_{MAX}$ então
5: se $|DF_2(c) - DF'_2(c \cup w_i)| \leq (\tau * FD_2(c))$ então
6: $candidate \leftarrow candidate \cup \{c, DF'_2(c \cup w_i), |DF_2(c) - DF'_2(c \cup w_i)|\}$
7: fim se
8: fim se
9: fim para
10: se ($candidate$) então
11: INSERTONMINIMAL($candidate$) ▷ Insere no DS_i no grupo com menor expan-
 são ou contração e atualiza a dimensão
 fractal do grupo

12: retorna verdadeiro
13: senão
14: retorna falso ▷ Nenhum grupo acomodou DS_i
15: fim se
16: fim função

```

---

algoritmo é encerrado sem sucesso na atualização do grupo corrente do fluxo de dados.

**Algoritmo 9 – Atualiza o grupo atual do Data Stream**


---

```

1: função UPDATEEFCC($(DS_i, x), C_j, \sigma_{MAX}, w_i, \tau$) ▷ Par (fluxo de dados, dimensão
 fractal); C_j grupo ao qual pertence
 o fluxo de dados; Desvio padrão
 máximo σ_{MAX} ; w_i conjunto de da-
 dos do fluxo de dados, τ parâme-
 tro de expansão ou contração dos
 grupos.

2: se $|x.DF_2 - C_j.DF_2| \leq \sigma_{MAX}$ então
3: se $|DF_2(C_j) - DF'_2(C_j \cup w_i)| \leq (\tau * DF_2(C_j))$ então
4: $C_j \leftarrow C_j \cup (DS_i, x, w_i)$
5: retorna verdadeiro
6: fim se
7: fim se
8: retorna falso
9: fim função

```

---

Tanto o algoritmo *BIG* quanto o *UpdateeFCC* são utilizados pelo módulo principal do eFCC; o algoritmo de mesmo nome (Alg.: 10) é o responsável por agrupar os fluxos de dados com comportamento e distribuição dos dados semelhantes, em um determinado período de tempo, em um conjunto disjunto  $\mathbb{C}$  de grupos.



**Algoritmo 10** – Agrupamento evolutivo de *Data Streams* (eFCC)

---

```

1: função EFCC($X, \mathbb{C}, \sigma_{MAX}, W, \tau$) $\triangleright$ Conjunto X contendo pares (DS_i, DF) , fluxo de dados
 e sua dimensão fractal, respectivamente; Partição de
 agrupamentos $\mathbb{C}$ tal que na primeira iteração $\mathbb{C} = \{\}$;
 Desvio padrão máximo σ_{MAX} ; W conjunto de janelas
 deslizantes de cada fluxo de dados, τ parâmetro de
 expansão ou contração dos grupos.

Saída: Partição $\mathbb{C}$ de agrupamentos com fluxos de dados com comportamento e distribuição
dos dados similares.

2: para todo $x \in X$ faça
3: se (DS_i, x_i) pertence a um grupo $C_j \in \mathbb{C}$ então
4: se UPDATEEFCC($C_j, x, \sigma_{MAX}, w_i, \tau$) = falso então
5: se BIG($(DS_i, x_i), \sigma_{MAX}, w_i, \tau$) = falso então
6: $\mathbb{C} \leftarrow \mathbb{C} \cup (C_{new} \leftarrow \text{CREATENEWCLUSTER}((DS_i, x_i), w_i))$
7: REARRANGETOCLOSEST(C_j, C_{new}, τ)
8: fim se
9: fim se
10: senão se BIG($(DS_i, x_i), \sigma_{MAX}, w_i, \tau$) = falso então
11: $C_{new} \leftarrow \text{CREATENEWCLUSTER}((DS_i, x_i), w_i)$
12: fim se
13: fim para
14: BRO($\mathbb{C}, W, \tau$)
15: OVERLAP($\mathbb{C}, W, \tau$)
16: retorna $\mathbb{C}$
17: fim função

```

---

Quando o algoritmo 10 inicia o seu processamento, não há nenhum grupo formado, ou seja,  $\mathbb{C} = \emptyset$ . Então, o algoritmo 10 inicia a iteração sobre o conjunto  $X$ , composto de pontos de dimensão fractal ( $df$ ) tal que cada ponto  $x$  é rotulado com seu respectivo fluxo de dados  $DS_i$  e, possui, também, o conjunto de pontos inserido na janela deslizante  $w_i$ . Primeiramente é verificado se o fluxo de dados  $DS_i$  pertence ou não a algum agrupamento  $C_i$  (Alg.: 10, linha 3), podendo ocorrer duas situações distintas: 1) o fluxo de dados ainda não foi agrupado e, 2) o fluxo de dados já foi agrupado em alguma iteração prévia.

**Situação 1):** caso o fluxos de dados  $DS_i$  já tenha sido inserido em algum grupo  $C_i$ , em alguma iteração anterior, então o eFCC tentará manter  $DS_i$  no mesmo grupo  $C_i$  se e somente se a dimensão fractal  $DF$  de  $DS_i$  for semelhante à dimensão fractal do grupo  $DF_2(C_i)$  e ao inserir os dados atuais de  $DS_i$  em  $C_i$  não ocorra uma expansão ou contração de  $DF_2(C_i)$  maior que a permitida (Alg.: 10, linha: 4). Se não for possível realizar a atualização do grupo corrente de  $DS_i$ , os grupos remanescentes em  $\mathbb{C}$  deverão ser verificados para acomodar  $DS_i$ , obedecendo os critérios estabelecidos no algoritmo 8 (Alg.: 10, linha: 5). Não encontrando nenhum grupo em  $\mathbb{C}$  que possa incorporar  $DS_i$  então um novo grupo  $C_{new}$  é criado e anexado a  $\mathbb{C}$  (Alg.: 10, linha: 6). Para cada um dos fluxos de dados restantes em  $C_i$  é verificado se eles permanecem em  $C_i$  ou se serão inseridos no novo grupo  $C_{new}$ , ou seja, cada um será novamente testado com relação aos



critérios de similaridade e distribuição dos dados (Alg.: 10, linha: 7).

**Situação 2):** supondo que  $DS_i$  ainda não foi agrupado (Alg.: 10, linha: 3) , então é necessário procurar por um grupo  $C \in \mathbb{C}$  para inserir  $DS_i$ , executando novamente a função BIG (Alg.: 10, linha 10). Não existindo nenhum grupo que possa receber  $DS_i$ , um novo grupo é criado e adicionado em  $\mathbb{C}$  (Alg.: 10, linha: 11)

Após processar todo  $X$ , o eFCC busca por fluxos de dados *outliers* (Alg.: 10, linha: 14) de dois modos: primeiro, buscando grupos cuja quantidade de  $(DS_i, x_i)$  inseridos seja uma pequena fração  $f$  de  $X$ , definida do mesmo modo que no método eFCDS; segundo, buscando fluxos de dados que se tornaram *outliers* em seus próprios grupos, devido aos processos de inserção de novos membros ou rearranjo. A Figura 28 apresenta a abordagem para detectar *outliers* dentro dos grupos. Primeiro é verificada a maior distância entre fluxos de dados do grupo, para identificar o diâmetro do grupo, conforme apresentado na Figura 28a. Então é analisado o diâmetro em relação ao desvio padrão do grupo, de acordo com o apresentado na Figura 28b. Por último são identificados os fluxos de dados, utilizando a Equação 5.4, que estão na faixa considerada anômala entre  $\mu_c + \sigma_c$  até diâmetro/2, onde  $\mu_{c_i}$  indica o centroide do grupo  $C_i$ ,  $\sigma_{c_i}$  o desvio padrão do grupo, conforme pode ser observado na Figura 28c.

$$DS_i(\text{outlier}) = \begin{cases} 1 & \text{se } DF_2(DS_i) > \mu_{c_i} + \sigma_{c_i} \\ 0 & \text{caso contrário} \end{cases} \quad (5.4)$$

Figura 28 – Processo para verificar os fluxos de dados que estão em dissonância com o grupo que estão inseridos.

Fonte: Elaborada pelo autor.

Os procedimentos para encontrar *Outliers* são apresentados em detalhes no algoritmo 11. Inicialmente são percorridos os grupos que possuem poucos fluxos de dados, de acordo com o parâmetro fornecido pelo usuário ( $f$ ). Caso esse parâmetro não seja fornecido, uma pequena fração da quantidade de fluxos de dados, definida empiricamente como 1% do total de fluxos de dados, caso esse valor seja maior que zero, senão é estabelecido que grupos com apenas um

fluxo de dados precisa ser testados para realocação (Alg.: 11, linha: 2). Após determinar  $f$ , são percorrido todos os grupos e aqueles que possuírem tamanho menor ou igual a  $f$  terão seus fluxos de dados testados para realocação (Alg.: 11, linha: 6), utilizando o algoritmo *BIG* (Alg.: 8). O segundo teste é realizado para encontrar os grupos que podem possuir *outliers* internamente (Alg.: 11, linha: 9). Se o grupo possui a possibilidade de ter *outliers*, os fluxos de dados alocados nele serão averiguados se estão fora do desvio padrão máximo (Alg.: 11, linha: 11) e, então, é avaliado se eles podem se encaixar em outro grupo (Alg.: 11, linha: 12), também utilizando o algoritmo *BIG* (Alg.: 8).

---

**Algoritmo 11** – Busca e reagrupa *Outliers* - BRO
 

---

```

1: função BRO($(\mathbb{C}, W, \sigma_{MAX}, \tau)$) ▷ Partição $\mathbb{C}$; W conjunto de Janelas desli-
 zantes; Desvio padrão máximo σ_{MAX} ; τ
 parâmetro de expansão ou contração dos
 grupos.
2: $f \leftarrow (f?f : ((\lceil |X|/100 \rceil) > 0 ? (\lceil |X|/100 \rceil) : 1))$
3: para todo $c \in \mathbb{C}$ faça
4: se $|c| \leq f$ então
5: para todo $DS \in c$ faça
6: $BIG((DS, DS.DF), \mathbb{C}, \sigma_{MAX}, DS.w, \tau)$
7: fim para
8: senão
9: se $(diam_c > (\mu_c + \sigma_{max}) * 2)$ então ▷ Testa diâmetro do grupo para identificar
 aqueles que possam possuir outliers
10: para todo $DS \in c$ faça
11: se $DS.DF - \mu_c > \sigma_{max}$ então
12: $BIG((DS, DS.x), \mathbb{C}, \sigma_{MAX}, DS.w, \tau)$
13: fim se
14: fim para
15: fim se
16: fim se
17: fim para
18: fim função

```

---

Retornando ao algoritmo 10, o último passo é verificar se existem grupos que estão em sobreposição (Alg.: 10, linha 15), unindo-os de modo semelhante ao algoritmo MERGEGROUP (Alg. 7) do método eFCDS, diferindo apenas na maneira que os fluxos de dados são inseridos nos grupos resultantes, pois deverão ser respeitadas as condições de comportamento e distribuição dos dados similares, característica fundamental do eFCC.

Após realizar uma iteração sobre o conjunto  $X$ , a partição  $\mathbb{C}$  contém grupos disjuntos com fluxos de dados que possuem comportamento e distribuição de dados similares entre si, em relação à dimensão fractal  $DF(DS_i)$  e  $DF(C_i)$ . E, a medida que novos dados chegam, o módulo de mineração é reexecutado, realizando novas operações de divisão, união e criação de novos grupos, fornecendo suporte incremental à evolução dos grupos ao longo do tempo, para retratar as mudanças de comportamento dos dados que surgirem ao longo do tempo.

Considerar, também, a distribuição dos dados para realizar o agrupamento de fluxos de dados impôs ao eFCC uma complexidade maior que o método eFCDS. A complexidade do eFCC ficou na ordem de  $O(e \times \eta_c \times E \times R \times |\mathbb{C}| + W \times |\mathbb{C}|)$ , onde  $W$  é o conjunto de janelas deslizantes.

## 5.3 Considerações Finais

Nesse Capítulo foram apresentados dois novos métodos para agrupamento de fluxos de dados multidimensionais, o eFCDS e o eFCC. Ambos os métodos tratam adequadamente a multidimensionalidade dos fluxos de dados, considerando a existência de correlações entre as dimensões, algo que os métodos do estado da arte ou não realizam, ou realizam de modo precário.

Conforme apresentado o eFCDS busca agrupar fluxos de dados com comportamento semelhantes ao longo do tempo, realizando uma única leitura dos dados e acompanhando as características evolutivas dos fluxos de dados, além de identificar fluxos de dados que tiveram comportamento anômalo em cada um dos períodos analisados. Sendo um método com complexidade baixa e tempo de processamento muito inferior aos concorrentes diretos da literatura, conforme será apresentado no Capítulo de Experimentos.

O eFCC, além das principais características do eFCDS, inova ao considerar a distribuição dos dados como critério para realizar a construção dos grupos, além de possuir uma abordagem mais complexa para detecção de *outliers*, verificando, inclusive, os fluxos de dados com comportamento anômalo dentro dos grupos já formados. Todavia, conforme apresentado, o eFCC possui complexidade computacional absoluta maior que o eFCDS.

No próximo Capítulo são apresentados os experimentos realizados para avaliar e validar o eFCDS e o eFCC, utilizando dados sintéticos e dados reais.



---

## EXPERIMENTOS

---

Nesse Capítulo serão apresentados os resultados de experimentos executados com dados reais e sintéticos para avaliar e validar os métodos eFCDS e eFCC propostos nesta tese de doutoramento para agrupamento de fluxos de dados multidimensionais. Com os experimentos formulados, buscar-se-a responder e validar as questões de pesquisas formuladas no [Capítulo 1 Introdução: Seção 1.3 Lacuna de Pesquisa e Objetivos do Trabalho](#).

Com o propósito de avaliar a qualidade dos métodos propostos em relação a coesão dos grupos formados e ao tempo de processamento, foi implementada a seguinte metodologia:

- Testar as abordagens desenvolvidas utilizando um conjunto de dados sintéticos.
- Testar os métodos utilizando um conjunto de dados reais, obtidos de sensores climáticos.
- Avaliar a qualidade dos grupos obtidos utilizando métricas de avaliação interna.
- Comparar o eFCDS e o eFCC com os métodos do estado da arte.
- Computar o tempo de processamento de todos os métodos avaliados.

Essa metodologia tem por objetivo avaliar os métodos propostos de acordo com os seguintes pontos chaves:

- Capacidade de agrupar corretamente fluxos de dados multidimensionais similares, considerando a existência de correlações entre dimensões.
- Coesão dos grupos gerados, ou seja, quão circunvizinhos os fluxos de dados de um grupo estão em relação ao centro do grupo. Com esse objetivo busca-se responder as questões de pesquisa 1 e 2, em que se verificará a influência de se correlacionar as dimensões e o uso da dimensão fractal na construção dos agrupamentos gerados.

- Qualidade da coesão dos grupos gerados frente à qualidade dos grupos obtidos pelos métodos concorrentes. Com esse objetivo busca-se corroborar as questões de pesquisa 1 e 2 respondidas pelo item anterior, demonstrando-se na prática que a coesão dos grupos gerados pode ser melhorada ao se utilizar a dimensão fractal.
- Capacidade dos métodos de acompanhar a evolução dos dados sem, contudo, degradar a qualidade e a coesão dos grupos obtidos.
- Capacidade de processar e agrupar dados em tempo real, utilizando-se de um conjunto de dados reais. Com esse objetivo busca-se responder a questão de pesquisa 3, verificando se o tempo de processamento dos fluxos de dados multidimensionais pode ser diminuindo ao realizar a construção dos agrupamentos utilizando-se a dimensão fractal.

Os resultados obtidos pelos métodos desenvolvidos, neste trabalho, foram comparados aos resultados dos métodos POD-Clus e ECM (ambos apresentados na Seção 3.1.4). Esses métodos, dentre os poucos propostos na literatura para agrupamento de fluxos de dados, foram escolhidos por também apresentarem soluções para alguns dos desafios tratados no eFCDS e eFCC, tais como: construir agrupamentos de qualidade com coesão entre os membros dos grupos, processamento rápido e incremental dos agrupamentos, representação compacta dos fluxos de dados, rastreabilidade das mudanças e identificação de *outliers*.

Conforme mencionado na Seção 3.1.4, desses dois métodos, apenas o POD-Clus trata a multidimensionalidade existente nos fluxos de dados. Todavia, ele apenas mensura a dissimilaridade entre as dimensões dos fluxos de dados, sem considerar as correlações que possam existir entre elas.

O ECM lida apenas com fluxos de dados unidimensionais, sendo assim, com o objetivo de proporcionar uma comparação mais igualitária e permitir que o ECM pudesse ser aplicado a fluxos de dados multidimensionais, foi calculada a dimensão fractal para cada sensor, assim, criaram-se fluxos de dados unidimensionais para cada sensor, que foram utilizados como entrada para a construção dos agrupamentos pelo ECM, tanto para os dados sintéticos, quanto para os dados reais.

Para garantir imparcialidade, todos os métodos avaliados utilizaram o mesmo computador pessoal com sistema operacional Ubuntu 16.04 LTS com 8GB de memória RAM, disco rígido de 1TB e processador Intel<sup>®</sup> Quad Core<sup>™</sup>i7-4790K, com velocidade de 4GHZ e 8MB de memória *cache*. E cada experimento foi executado 10 vezes a fim de obter uma média dos resultados obtidos. Todos os métodos, inclusive o ECM (WIDIPUTRA; PEARS; KASABOV, 2011) e o POD-Clus (CHAOVALIT, 2012), foram implementados utilizando a linguagem C++ utilizando as especificações fornecidas nas publicações de cada um.

O restante desse Capítulo está organizado da seguinte maneira: na Seção 6.1 são apresentados o processo de construção de cada um dos conjuntos de fluxos de dados sintéticos,

assim como, os resultados obtidos pelos métodos eFCDS, eFCC e os métodos competidores ao agrupar esses tipos de fluxos de dados; na Seção 6.2 são apresentados o conjunto de dados reais, juntamente com suas características, e os resultados obtidos por cada um dos métodos; e na Seção 6.3 as considerações finais sobre os resultados obtidos.

## 6.1 Dados Sintéticos

Nessa seção, são apresentados os conjuntos de fluxos de dados construídos sinteticamente, com o objetivo de avaliar os métodos desenvolvidos nessa tese de doutorado e os principais métodos concorrentes. Esse conjunto de fluxo de dados contribuirá para avaliar os efeitos que as variações dos parâmetros ocasionam em cada um dos métodos. Sendo possível avaliar a qualidade dos agrupamentos obtidos e a capacidade dos métodos em agrupar fluxos de dados similares identificando as correlações entre as dimensões. Com esses objetivos em mente, foram criados fluxos de dados tridimensionais sintéticos, com diferentes distribuições e formas, conforme as Fórmulas apresentadas nas Equações 6.1. Os dados gerados por essas equações são: dados com formato de uma *borboleta* (Equação 6.1), chamado apenas de *Borboleta*, dados *uniformemente distribuídos* em um espaço tridimensional (Equação 6.2), chamado de *UnifDist*, dados formando metade de uma *esfera* (Equação 6.3), chamado de *semi-esfera*, e dados formando *retas* paralelas (Equação 6.4), chamado de *retas*. No total foram gerados 10 fluxos de dados de cada fórmula, totalizando 40 fluxos de dados cada um com três dimensões e 365 pontos.

$$Borb(x,y,z) = \begin{cases} x = 5 + r * \sin(\pi * i * 180/nptos) * \cos(\pi * i * 180/nptos/2) \\ y = 5 + r * \sin(\pi * i * 180/nptos) * \sin(\pi * i * 180/nptos/2) \\ z = a + i/nptos \end{cases} \quad (6.1)$$

$$onde : nptos = 365; i = [0, nptos); a = [0, 10); r = 1$$

$$Unif(x,y,z) = \begin{cases} x = 2 + a + 2 * \sin(nrand_1/2) * \cos(nrand_4) \\ y = 2 + a + 2 * \sin(nrand_2/2) * \sin(nrand_5) \\ z = 2 + a + 2 * \cos(nrand_3/2); \end{cases} \quad (6.2)$$

$$onde : nptos = 365; i = [0, nptos); a = [0, 10); nrando_x = randon([0, 2\pi])$$

$$SemiEsf(x,y,z) = \begin{cases} x = a + 2 * \sin(\pi * i/npts/2) * \cos(\pi * i * 360/npts) \\ y = a + 2 * \sin(\pi * i/npts/2) * \sin(\pi * i * 360/npts) \\ z = a + 2 * \cos(\pi * i/npts/2) \end{cases} \quad (6.3)$$

$$onde : nptos = 365; i = [0, nptos); a = [0, 10)$$

$$RL(x,y,z) = \begin{cases} x \\ y = i/nptos * xmax \\ z = (x + 2 * y + 5) / (xmax + 2 * nptos + 5) * xmax \end{cases} \quad (6.4)$$

$$onde : x = [0, 10); nptos = 365; i = [0, nptos); xmax = 10$$

Quadro 2 – Sumarização das características dos conjuntos sintéticos

| Tipo de conjunto   | Nro fluxos | Diferenciação entre fluxos                                                                                                                                                                                                                                              | Valores de $\mathcal{D}_2$ | Figura de exemplo |
|--------------------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------|-------------------|
| <i>Borboleta</i>   | 10         | Centro de cada Borboleta $x = y = 5$ , e variando o eixo $z$ em uma unidade para cada Borboleta                                                                                                                                                                         | 1,5944                     | Figura 29(a)      |
| <i>UnifDist</i>    | 10         | Números randômicos são gerados ao redor de cada centro, da UnifDist0 $x = y = z = 2$ , da UnifDist1 $x = y = z = 3, \dots$ , da UnifDist9 $x = y = z = 11$                                                                                                              | de 2,6973 até 2,9074       | Figura 29(b)      |
| <i>semi-esfera</i> | 10         | Para cada fluxo ocorre a variação de uma unidade nos eixos $x$ , $y$ e $z$ . Assim, a <i>semi-esfera</i> 0 iniciará na posição $x = y = z = 0$ , a <i>semi-esfera</i> 1 na posição $x = y = z = 1$ , e assim sucessivamente até <i>semi-esfera</i> 9, $x = y = z = 9$ . | 2,0456                     | Figura 29(c)      |
| <i>Reta</i>        | 10         | As <i>reta</i> estão posicionada no eixo $x$ , em intervalos de uma unidade, iniciando no $x = 0$                                                                                                                                                                       | de 1,0609 até 1,0943       | Figura 29(d)      |

A sumarização das características de cada um dos conjuntos gerados é apresentada no Quadro 2.

As formas definidas para os conjuntos de dados permitem ilustrar diferentes tipos de correlação entre dimensões, de correlações lineares (conjunto reta) a dimensões completamente independentes (conjunto uniforme), passando por correlações mais complexas como nos conjuntos borboleta e semiesfera. A representação gráfica de cada um dos conjuntos é apresentada na Figura 29.

A Figura 30 apresenta um exemplo de cada um dos conjuntos gerados pelas equações



Figura 29 – Conjuntos sintéticos

(a) Conjunto de retas sintéticas gerados pela Equação 6.1

(b) Conjunto de distribuições uniformes (*Unif-Dist*) sintéticas gerados pela Equação 6.2 (c) Conjunto de *semi-esferas* sintéticas geradas pela Equação 6.3

(d) Conjunto de retas sintéticas gerados pela Equação 6.4

Fonte: Elaborada pelo autor.

em uma mesma representação gráfica.

Para analisar os agrupamentos gerados e o impacto da variação dos parâmetros de cada método, foram considerados 4 cenários distintos e relevantes para a construção dos agrupamentos:

1. Cada fluxo de dados ser atribuído a um grupo distinto, mediante o uso de valores de limiar muito baixos.
2. Formar apenas um grupo com todos os fluxos de dados, utilizando limiares muito altos.
3. Não agrupar fluxos de dados diferentes em um mesmo grupo, independente da quantidade de grupos formados.

Figura 30 – Conjunto de fluxos de dados sintéticos gerados pelas Equações 6.1 até 6.4.

Fonte: Elaborada pelo autor.

4. Formar exatamente um grupo para cada uma das 4 formas, agrupando os seus respectivos fluxos de dados.

O primeiro cenário permite avaliar quão granular são os grupos obtidos utilizando-se parâmetros de limiar muito pequenos (considerando a semântica dos limiares adotados nos métodos avaliados). Nesse caso, espera-se que apenas valores muito baixos dos limiares separem completamente os fluxos de dados, gerando grupos individuais. Vale ressaltar que ao atender os requisitos do primeiro cenário, automaticamente os requisitos do terceiro são cumpridos também, pois se é formado um grupo para cada um dos 40 fluxos de dados, não serão inseridos fluxos distintos em um mesmo grupo. Entretanto, esse não é o objetivo do terceiro cenário, mas sim avaliar se ao utilizar parâmetros com valores intermediários, a homogeneidade dos grupos formados se mantém para alguns desses parâmetros, reduzindo a quantidade de fluxos de dados distintos inseridos num mesmo grupo. Nesse caso, o esperado é que apenas valores de limiar mais altos resultem em apenas um grupo, ou seja, não consigam diferenciar os fluxos de dados de acordo com os critérios utilizados pelos métodos. O segundo cenário é utilizado para avaliar o oposto do primeiro, ou seja, quais são os valores de limiar a partir do qual todos os fluxos ficam em um único grupo. E o último cenário, o mais importante de todos, permite avaliar a capacidade dos métodos de encontrar 4 grupos representando cada uma das formas geradas pelas Equações 6.1 até 6.4, ou seja, o agrupamento ideal. Ao atender a esse cenário o método também atenderá o terceiro concomitantemente.

Traçados os objetivos e resultados esperados, o passo seguinte foi a configuração dos parâmetros de cada um dos métodos. Um parâmetro comum a todos os métodos é o tamanho da janela deslizante, definido com 365 itens de dados e sem deslocamento, para englobar todo o conjunto de dados em uma única janela.

Conforme mencionado na Seção 3.1.4.4, o ECM baseia-se em um conjunto de regras de decisão e um parâmetro de limiar para decidir como criar, remover ou unir os grupos. Portanto, para testar o ECM utilizou-se como parâmetro de limiar os valores [0,001; 0,01; 0,1; 0,2; 0,3; 0,4; 0,5; 0,6; 0,7; 0,8; 0,9; 1,0; 2,0; 3,0; 4,0; 5,0; 10,0]. O Quadro 3, apresenta a quantidade de grupos obtidos pelo ECM em cada uma das configurações, quantos grupos foram formados com mais de um fluxo de dados inserido, quantos grupos inseriram formas dissimilares e, quais dos 4 cenários foram atendidos (indicado por ✓) com cada um dos parâmetros.

Quadro 3 – Resultado obtido pelo ECM variando o parâmetro de limiar

| Valor Limiar | Nro Grupos | Nro Grupos 1+ | Qtde Grupos Dissimilar | Cen. 1 | Cen. 2 | Cen. 3 | Cen. 4 |
|--------------|------------|---------------|------------------------|--------|--------|--------|--------|
| 0,001        | 40         | 0             | 0                      | ✓      |        |        |        |
| 0,01         | 40         | 0             | 0                      | ✓      |        |        |        |
| 0,1          | 38         | 2             | 1                      |        |        |        |        |
| 0,2          | 31         | 8             | 6                      |        |        |        |        |
| 0,3          | 31         | 8             | 6                      |        |        |        |        |
| 0,4          | 30         | 9             | 8                      |        |        |        |        |
| 0,5          | 30         | 9             | 8                      |        |        |        |        |
| 0,6          | 20         | 9             | 9                      |        |        |        |        |
| 0,7          | 11         | 7             | 7                      |        |        |        |        |
| 0,8          | 2          | 2             | 1                      |        |        |        |        |
| ⋮            | 2          | 2             | 1                      |        |        |        |        |
| 10,0         | 2          | 2             | 1                      |        |        |        |        |

Os resultados apresentados pelo ECM indicam que esse método não possui a habilidade para criar grupos homogêneos, pois para quase todos os parâmetros avaliados, com exceção dos limiares muito baixos que não possuem grupos com mais de um fluxo de dados, o ECM criou grupos mesclados. Por exemplo, no valor de limiar 0,6 foram criados 20 grupos, mas somente 9 grupos foram criados com mais de um fluxo de dados inserido e desses, 9 apresentaram grupos com fluxos de dados dissimilares, ou seja, 100% dos grupos com mais de um fluxo de dados apresentou fluxos de dados dissimilares. O mesmo problema ocorreu para o parâmetro 0,7. Já os valores maiores que 0,7 criaram dois grupos apenas, um para os fluxos de dados *UnifDist* e outro para os outros fluxos de dados.

O PODClus, conforme apresentado na Seção 3.1.4.3, requer a configuração de três parâmetros: o número inicial de grupos, o limiar de dissimilaridade entre um fluxo de dados e o grupo mais próximo e a quantidade mínima de fluxos de dados para um grupo de *outliers* se

transformar em um grupo normal. Devido ao conjunto sintético possuir 4 tipos de formas bem distintas, o número inicial de grupos foi configurado para 4. Já o limiar de dissimilaridade ( $\sigma$ ) variou de 0,001 até 1,0. E a quantidade de fluxos de dados necessárias para um grupo deixar de ser *outlier* foi configurado para três fluxos, valor definido empiricamente em 10% do recomendado pelo autor do POD-Clus. Os resultados são apresentados no Quadro 4, onde a primeira coluna é o limiar de dissimilaridade, a segunda a quantidade de grupos *outliers*, a terceira representa o número de grupos formados, a quarta a quantidade de grupo com fluxos dissimilar, a quantidade de grupos com mais de um fluxo de dados e as outras colunas representam os cenários. É possível

Quadro 4 – Resultado obtido pelo PODClus variando o parâmetros de limiar de similaridade.

| $\sigma$ | Qtde Outlier | Nro Grupos | Qtde Grupos Dissimil | Qtde 1+ | Cen. 1 | Cen. 2 | Cen. 3 | Cen. 4 |
|----------|--------------|------------|----------------------|---------|--------|--------|--------|--------|
| 0,001    | 10           | 15         | 6                    | 9       |        |        |        |        |
| 0,01     | 10           | 12         | 6                    | 8       |        |        |        |        |
| 0,1      | 8            | 11         | 5                    | 6       |        |        |        |        |
| 0,2      | 7            | 5          | 2                    | 2       |        |        |        |        |
| 0,3      | 7            | 3          | 1                    | 1       |        |        |        |        |
| 0,4      | 5            | 2          | 1                    | 1       |        |        |        |        |
| 0,5      | 4            | 1          | 1                    | 1       |        |        |        |        |
| 1,0      | 0            | 1          | 1                    | 1       |        | ✓      |        |        |

notar que o POD-Clus também não consegue gerar grupos de dados sem que exista fluxos dissimilares. Mesmo para valores muito pequenos, o POD-Clus já criar grupos de fluxos de dados com mais de uma forma inseridos neles. A partir do valor de dissimilaridade 0,2, 100% dos grupos com mais de um fluxo de dados apresentaram fluxos de dados dissimilares. Tendo sido atendido apenas o cenário 2, para valores de dissimilaridade maior ou igual a 1, o POD-Clus demonstra que não consegue gerar grupos de qualidade tão facilmente. Por ser baseado apenas em medidas descritivas, como média, desvio padrão, matriz de covariância e regressão linear, tornam o POD-Clus limitado quanto à descrição dos grupos formados que ele pode representar.

Para o eFCDS, desenvolvido nesse doutoramento e apresentado na Seção 5.1, é preciso variar apenas o parâmetro de máximo desvio padrão ( $\sigma_{max}$ ), utilizado para determinar quão próximos devem estar os fluxos de dados para serem inseridos em um mesmo grupo. Os resultados dessa variação de parâmetro são apresentados no Quadro 5, vale ressaltar que entre os valores 0,5 e 1,0, ocultos no quadro, os valores obtidos foram os mesmos do limiar 0,5 e valores maiores ou iguais a 1,1 geram apenas um agrupamento.

Conforme é possível constatar, o único cenário não atendido pelo eFCDS foi o primeiro. A principal razão do método não ter criado um grupo para cada um dos 40 fluxos de dados, mesmo com valores de limiar muito baixos, é o fato de todos os 10 fluxos em forma de borboleta (Equação 6.1) possuírem  $\mathcal{D}_2 = 1.59442$ , apesar do deslocamento no eixo-z existente entre eles;

Quadro 5 – Resultado obtido pelo eFCDS variando o parâmetro de máximo desvio padrão ( $\sigma_{max}$ )

| $\sigma_{max}$ | Nro Grupos | Cen. 1 | Cen. 2 | Cen. 3 | Cen. 4 |
|----------------|------------|--------|--------|--------|--------|
| 0,001          | 19         |        |        |        |        |
| 0,01           | 11         |        |        | ✓      |        |
| 0,05           | 6          |        |        | ✓      |        |
| 0,1            | 5          |        |        | ✓      |        |
| 0,2            | 4          |        |        | ✓      | ✓      |
| 0,3            | 4          |        |        | ✓      | ✓      |
| 0,4            | 4          |        |        | ✓      | ✓      |
| 0,5            | 3          |        |        |        |        |
| 1,0            | 3          |        |        |        |        |
| 1,1            | 1          |        | ✓      |        |        |

o mesmo ocorre para os fluxos em forma de semiesfera (Equação 6.3), com  $\mathcal{D}_2 = 2.04561$ , apesar do deslocamento em uma unidade em cada um dos eixos (x, y e z). No entanto, esse comportamento era esperado, uma vez que o eFCDS considera apenas as correlações existentes entre as dimensões dos fluxos de dados como critério de agrupamento. Como explicado na Seção 5.1, aspectos como intervalos de valores e distribuição dos dados não são o foco da abordagem proposta no eFCDS.

Todos os demais cenários foram atendidos para algum valor de limiar testado. Como esperado, apenas um valor de desvio padrão alto gerou um único grupo para todos os fluxos de dados. Valores intermediários, mais baixos, resultaram na geração de grupos homogêneos, sem mistura de formas, o que é um resultado relevante (Cenário 3). Além disso, em uma das configurações de parâmetro ( $\sigma_{max} = 0,2$ ), o eFCDS gerou o agrupamento ideal (Cenário 4), agrupando fielmente os fluxos de dados dos 4 tipos de forma, cada um em seu respectivo grupo.

O eFCC, também desenvolvido nesse doutoramento e apresentado na Seção 5.2, requer dois parâmetros para construir seus agrupamentos. O primeiro é o mesmo parâmetro já apresentado nos experimentos com o eFCDS, o máximo desvio padrão ( $\sigma_{max}$ ). O segundo, que auxilia na identificação dos fluxos de dados com distribuição de dados similar, é o parâmetro de máxima contração ou expansão de um grupo ( $\tau$ ). Para realizar a variação dos parâmetros utilizou-se a seguinte metodologia: 1) variou-se o  $\sigma_{max}$  e manteve-se o  $\tau$  fixo enquanto não se registrou mudanças nos agrupamentos; 2) registrando-se a mudança, manteve-se o valor de  $\sigma_{max}$  e variou-se o valor de  $\tau$ ; 3) quando o valor de uma iteração para a outra mantinha-se, os resultados obtidos não eram registrados, por exemplo para os valores  $\sigma_{max} = 0,1$  e  $\tau = 0,3; 0,4e0,5$  os resultados foram os mesmos que para os valores  $\sigma_{max} = 0,1$  e  $\tau = 0,2$ . Os resultados desse minucioso conjunto de avaliações são apresentados no Quadro 6.

É possível notar que o eFCC consegue, a partir da configuração de parâmetros, atender aos requisitos dos 4 cenários propostos. Ao contrário do eFCDS, o eFCC separou completamente os 40 fluxos de dados, para os valores mínimos dos parâmetros. Isso ocorreu porque, além das correlações entre dimensões, a distribuição dos dados e os intervalos de valores das dimensões

Quadro 6 – Resultado obtido pelo eFCC variando os parâmetros máximo desvio padrão ( $\sigma_{max}$ ) e limiar de expansão ou contração  $\tau$ .

| $\sigma_{max}$ | $\tau$ | Nro Grupos | Cen. 1 | Cen. 2 | Cen. 3 | Cen. 4 |
|----------------|--------|------------|--------|--------|--------|--------|
| 0,005          | 0,005  | 40         | ✓      |        |        |        |
| 0,01           | 0,01   | 36         |        |        | ✓      |        |
| 0,05           | 0,005  | 37         |        |        | ✓      |        |
| 0,05           | 0,01   | 36         |        |        | ✓      |        |
| 0,05           | 0,1    | 18         |        |        | ✓      |        |
| 0,1            | 0,005  | 37         |        |        | ✓      |        |
| 0,1            | 0,01   | 36         |        |        | ✓      |        |
| 0,1            | 0,1    | 18         |        |        | ✓      |        |
| 0,1            | 0,2    | 10         |        |        | ✓      |        |
| 0,1            | 0,6    | 7          |        |        | ✓      |        |
| 0,1            | 1      | 5          |        |        | ✓      |        |
| 0,2            | 0,1    | 17         |        |        | ✓      |        |
| 0,2            | 0,2    | 9          |        |        | ✓      |        |
| 0,2            | 0,6    | 4          |        |        | ✓      | ✓      |
| 0,2            | 1      | 4          |        |        | ✓      | ✓      |
| 0,5            | 0,1    | 14         |        |        | ✓      |        |
| 0,5            | 0,5    | 4          |        |        |        |        |
| 0,5            | 1      | 3          |        |        |        |        |
| 1              | 0,1    | 8          |        |        | ✓      |        |
| 1              | 0,5    | 2          |        |        |        |        |
| 1              | 1      | 2          |        |        |        |        |
| 1,1            | 1      | 1          |        | ✓      |        |        |

também são critérios de agrupamento na abordagem proposta no eFCC. Logo, o deslocamento dos dados no conjunto de retas, por exemplo, agora é levado em consideração no processo de agrupamento. Vale ressaltar ainda que o eFCC conseguiu gerar grupos homogêneos a partir uma variedade maior de valores de limiar, indicando menor sensibilidade em relação aos parâmetros.

Quadro 7 – Sumarização dos resultados obtidos pelos métodos, para cada uma das situações apresentadas

| Metodos | Cen. 1 | Cen. 2 | Cen. 3 | Cen. 4 |
|---------|--------|--------|--------|--------|
| ECM     | ✓      |        |        |        |
| PODClus |        | ✓      |        |        |
| eFCDS   |        | ✓      | ✓      | ✓      |
| eFCC    | ✓      | ✓      | ✓      | ✓      |

Dessa maneira, o Quadro 7, apresenta os resultados obtidos por cada um dos métodos de maneira sumarizada. Evidenciando que tanto o eFCDS, quanto o eFCC são capazes de gerar grupos de qualidade, agrupando fluxos de dados similares e identificando as correlações entre eles.

## 6.2 Dados Reais

O conjunto de dados reais é composto por dados coletados por 288 sensores climáticos localizados em diferentes regiões do Brasil. Cada sensor coleta diariamente 4 medidas distintas: temperatura mínima, temperatura máxima, precipitação e radiação solar. Essas medidas foram coletadas no período de janeiro de 1960 a dezembro de 2010, representando 21024000 medidas geradas pelos sensores nesse período. Esse conjunto de dados foi fornecido por pesquisadores da EMBRAPA de Campinas e do CEPAGRI da Unicamp, parceiros em projetos de pesquisa do Grupo de Bases de Dados e Imagens (GBdI), do Instituto de Ciências Matemáticas e de Computação (ICMC). Essa colaboração motivou a escolha desse conjunto de dados reais para estudo experimental neste trabalho de doutoramento.

Nos experimentos realizados neste trabalho, cada sensor climático corresponde a um fluxo de dados em que cada item de dados é composto por 4 dimensões (temperaturas máxima e mínima, radiação solar e precipitação). Então, para realizar os testes comparativos e verificar a performance utilizando o conjunto de dados reais, cada um dos métodos foi executado com diversas configurações de parâmetros. Os parâmetros que produziram os melhores resultados, para todos os métodos avaliados, são os especificados a seguir. Para o POD-Clus, a janela deslizante foi configurada para possuir tamanho total de 1825 medidas, com deslocamento de 365, representando 1 ano de dados; um grupo torna-se *outlier* se possuir menos de três fluxos de dados, para unir dois grupos o limiar foi de 0,5; e para todos os outros parâmetros foram utilizados os valores padrão. Os parâmetros do eFCDS foram ajustados conforme segue: ao máximo desvio padrão ( $\sigma_{max}$ ) foi atribuído o valor de 0,05, a janela deslizante foi configurada em 5 períodos de contagem, cada um contendo 365 eventos, perfazendo o tamanho máximo de 1825, de modo que uma janela deslizante gerasse um valor de dimensão fractal. O eFCC utilizou o mesmo valor de máximo desvio padrão ( $\sigma_{max}$ ) e de janela deslizante que o eFCDS. O parâmetro de  $\tau$  de expansão e contração permite que a dimensão fractal de um grupo cresça ou diminua em 20% o seu tamanho. Para o ECM, o limiar de correlação foi definido como 0,6 e a dimensão fractal foi obtida do mesmo modo que o eFCDS.

Efetuada as configurações dos parâmetros de cada um dos métodos, a primeira avaliação realizada foi a capacidade dos métodos em acompanhar a evolução dos dados ao longo do tempo, refletida na formação, exclusão ou união dos grupos. Esses resultados são apresentados na Figura 31. É possível notar na Figura 31 que todos os métodos buscam refletir o comportamento dos dados, criando e excluindo grupos de iteração para iteração. Nessa Figura também é possível notar que tanto o eFCDS e o eFCC encontraram mais grupos que seus concorrentes no conjunto de dados. Isso se deu devido ao fato de ambos os métodos considerarem o comportamento dos dados para formarem os grupos, sendo mais restritivos na formação dos grupos que seus concorrentes. Além disso, o eFCC também considera a distribuição dos dados como critério de agrupamento, aumentando, desse modo, a quantidade de grupos que possuem distribuição de dados distintas.

Figura 31 – Evolução dos grupos formados em cada uma das iterações.

Fonte: Elaborada pelo autor.

A segunda etapa de avaliação foi verificar a coesão de cada grupo. Para isso calculou-se qual porcentagem o desvio padrão representava em relação aos valores médios dos grupos em cada uma das iterações, para cada uma das dimensões. Esses resultados são apresentados na Figura 32.

Figura 32 – Resultados comparativos das porcentagens de desvio padrão dos grupos formados pelos métodos

(a) Média da porcentagem do desvio padrão para cada iteração dos métodos concorrentes

(b) Resultado comparativo da média da porcentagem do desvio padrão do eFCDS e do eFCC.

Fonte: Elaborada pelo autor.

A Figura 32a apresenta a porcentagem média de desvio padrão para o ECM e para todas as dimensões dos fluxos de dados nos agrupamentos gerados pelo POD-Clus, em relação aos valores médios dos dados em cada iteração e em cada dimensão (temperatura mínima - TMin, temperatura máxima - TMax, radiação solar - Srad e precipitação - Prec). A Figura 32b mostra



o resultado da média de desvio padrão entre o eFCDS e o eFCC em relação aos valores das dimensões fractais dos fluxos de dados.

Dessa maneira, é possível observar, na Figura 32a, que a porcentagem do desvio padrão, obtida pelos métodos concorrentes, foi muito superior aos valores auferidos pelo eFCDS e pelo eFCC, que ficaram próximos de zero, indicando uma elevada coesão na formação dos grupos em ambos os métodos desenvolvidos (Figura 32b).

A coesão dos grupos obtidos pelo ECM variaram de  $\approx 7,7\%$  a  $\approx 12\%$ , do POD-Clus variaram de  $\approx 13,4\%$  até  $\approx 18,7\%$  para a temperatura mínima,  $\approx 16,9\%$  até  $\approx 27,1\%$  para a temperatura máxima,  $\approx 34,7\%$  até  $\approx 45,9\%$  para a radiação solar e  $\approx 247,6\%$  até  $\approx 268,4\%$  para a precipitação. Também é possível notar que o desvio padrão do POD-Clus para a precipitação foi quase 2,5 vezes maior que a média dos valores da dimensão. O alto valor do desvio padrão ocorre por causa da característica de o Brasil possuir períodos de chuva bem definidos, em que a precipitação varia de 1200 até 1500 milímetros por ano. Como o POD-Clus considera cada dimensão individualmente, conforme apresentado na Equação 3.7, então, quanto maior o valor do desvio padrão de uma dimensão, menor é sua influência na medida de distância, afetando a qualidade dos grupos resultantes.

Figura 33 – Resultado de Silhueta do eFCDS.

Fonte: Elaborada pelo autor.

Conforme mencionado anteriormente na Seção 3.2, há diversos métodos para mensurar a qualidade dos resultados gerados por métodos de agrupamento. Nesta tese, em razão do conjunto de dados reais de interesse, devido a característica desse conjunto de dados não possuir um conjunto de validação, tampouco rótulos associados aos fluxos de dados, tornou-se imprescindível utilizar uma medida interna de avaliação de qualidade. Assim, optou-se pela medida da Silhueta, bastante utilizada em trabalhos correlatos por conseguir representar quão bem agrupados estão os fluxos de dados em seus próprios grupos em relação aos demais. Essa medida, conforme apresentado na Seção 3.2.1.2, procura avaliar quão bem dispostos estão os elementos dentro do grupo que se encontram em relação aos outros grupos existentes. Quando um fluxo de dados está corretamente agrupado o valor de Silhueta estará acima de 0,5, chegando ao máximo de um. Quando esse fluxo de dados deveria estar agrupado em outro grupo o valor de Silhueta fica entre

-1 e zero. Valores de Silhueta entre zero e 0,5 indicam que o fluxo de dados está corretamente agrupado, embora esteja relativamente próximo a outro grupo. Então, quanto maior o valor da Silhueta de um grupo (geralmente com valores acima de 0,5) mais coeso está o grupo em análise.

As medidas de qualidade obtidas para os agrupamentos gerados pelos métodos são apresentadas nas Figuras 33a até 36h. Nessas figuras, cada grupo está indicado por uma cor diferente e há um espaço entre cada grupo para facilitar a visualização.

As Figuras 33a e 33b apresentam o resultado da Silhueta dos grupos formados pelo eFCDS na primeira e na última iteração, respectivamente. O eFCDS construiu grupos com 68% da medida de Silhueta dos fluxos de dados acima de 0,5, indicando que a maioria dos fluxos de dados estão apropriadamente agrupados e apenas 4,2% deveriam estar em um grupo diferente na primeira iteração e 7,6% na última.

Figura 34 – Resultado de Silhueta do eFCC.

Fonte: Elaborada pelo autor.

Nas Figuras 34a e 34b são exibidas as medidas de Silhueta dos grupos formados pelo eFCC para a primeira e última iteração, respectivamente. Embora inicialmente o eFCC tenha apresentado um resultado ligeiramente inferior ao eFCDS para a medida de Silhueta, com a evolução dos grupos a cada iteração, o eFCC construiu grupos com 75,5% da medida de Silhueta dos fluxos de dados acima de 0,5, representando uma melhora de aproximadamente 11% em relação ao eFCDS.

As Figuras 35a e 35b exibem a medida de Silhueta calculada para os grupos formados pelo método ECM na primeira e última iteração, respectivamente. Nessa análise, apenas três fluxos de dados foram corretamente agrupados na última iteração, com valor de Silhueta acima de 0,5, a baixa qualidade dos grupos gerados pelo ECM ocorre porque o método baseia a construção dos agrupamentos em regras simples e porque a RMONC, utilizada pelo ECM, encontra apenas correlações lineares.

As Figuras 36a até 36h explicitam o resultado da Silhueta dos grupos construídos utilizando o POD-Clus na primeira e na última iteração, nessa ordem, para as medidas de radiação solar, temperatura mínima, temperatura máxima e precipitação. Como mencionado

Figura 35 – Resultado da Silhueta obtida pelo ECM para os agrupamentos da primeira e última iteração

Fonte: Elaborada pelo autor.

anteriormente, o POD-Clus trabalha com cada dimensão independentemente e essa abordagem contribui para a qualidade inferior dos agrupamentos conforme aponta a medida da Silhueta para cada uma das dimensões. O melhor resultado obtido pelo POD-Clus foi na dimensão da temperatura máxima da última iteração, com 5% das medidas de Silhueta acima de 0,5. Se forem considerados todos os resultados de Silhueta acima de zero, então o melhor resultado foi obtido na dimensão da temperatura mínima, também na última iteração, com mais de 20% dos valores acima de zero.

Comparando os resultados apresentados na Figura 32a com os resultados exibidos nas Figuras 33a até 36h, é plausível concluir que os agrupamentos formados pelo eFCDS são até 93,5% melhores que os agrupamentos formados pelo ECM e até 99,9% melhores que os gerados pelo POD-Clus, quando avaliamos a coesão dos grupos formados. Sendo os grupos gerados pelo eFCDS 14 vezes mais compactos que os gerados pelo ECM e aproximadamente 20 vezes mais compactos que aqueles que o POD-Clus gerou. Isso indica que ao utilizar a medida da dimensão fractal para representar os dados e considerar a correlação existente entre as dimensões agrupar fluxos de dados multidimensionais, a formação alcançada é de melhor qualidade do que utilizando-se apenas medidas descritivas e correlações lineares.

O eFCC, além de agrupar fluxos de dados com comportamento similar ao longo do tempo, busca também agrupar aqueles com distribuição similar dos dados. Para ilustrar o efeito desse critério no processo de agrupamento, foi escolhida, casualmente, uma iteração e um fluxo de dados, chamado de  $a$  para efeitos de explanação, contido em um grupo qualquer  $G$ . Então, do agrupamento de  $a$  outro fluxo de dados ( $b$ ) foi escolhido aleatoriamente. Assim sendo, para efeito comparativo, buscou-se na mesma iteração do eFCDS, no grupo em que se encontra  $a$ , chamado  $H_a$ , outro fluxo de dados  $c$ , tal que  $b \notin H_a$ . Então plotou-se os dados  $[a, b]$  e  $[a, c]$ , das três primeiras dimensões (temperatura mínima, temperatura máxima e radiação solar), para verificar a distribuição dos dados. Esse resultado pode ser conferido na Figura 37. É possível observar que os dados estão melhor distribuídos no grupo formado pelo método eFCC (Figura 37a), do que no grupo formado pelo eFCDS (Figura 37b). Para ampliar essa percepção, foram escolhidos

Figura 36 – Resultados do cálculo da Silhueta dos agrupamentos obtidos pelo POD-Clus para cada uma das dimensões dos fluxos de dados

(a) Cálculo da Silhueta da **radiação solar** para a **primeira** iteração.

(b) Cálculo da Silhueta da **radiação solar** para a **última** iteração.

(c) Cálculo da Silhueta da **temperatura mínima** para a **primeira** iteração.

(d) Cálculo da Silhueta da **temperatura mínima** para a **última** iteração.

(e) Cálculo da Silhueta da **temperatura máxima** para a **primeira** iteração.

(f) Cálculo da Silhueta da **temperatura máxima** para a **última** iteração.

(g) Cálculo da Silhueta da **precipitação** para a **primeira** iteração.

(h) Cálculo da Silhueta da **precipitação** para a **última** iteração.

Fonte: Elaborada pelo autor.

outros dois fluxos de dados, diferentes de  $b$  e  $c$ , pertencentes a  $G$  e  $H_a$  respectivamente, e o resultado comparativo é apresentado na Figura 38. Observa-se o mesmo comportamento na distribuição dos dados na Figura 37 e 38, onde os dados dos fluxos de dados estão uniformemente distribuídos, sem lacunas entre os dois conjuntos de dados.

Para verificar a capacidade dos métodos em lidar com fluxos de dados reais e que muitas vezes produzem dados a elevadas taxas, foi calculado o tempo de processamento de

Figura 37 – Resultado comparativo da distribuição dos dados nos grupos entre o eFCDS e eFCC, grupo de Juíz de Fora - MG (JDFO).

Fonte: Elaborada pelo autor.

Figura 38 – Resultado comparativo da distribuição dos dados nos grupos entre o eFCDS e eFCC, comparando outras estações do grupo de Juíz de Fora - MG (JDFO).

Fonte: Elaborada pelo autor.

cada um dos métodos. Os resultados obtidos são apresentados na Figura 39. Nessa Figura são apresentados os tempos de processamento gastos por cada um dos métodos para a construção de seus agrupamentos. É possível apurar que o eFCDS construiu seus grupos gastando 1/3 do tempo utilizado pelo ECM, aproximadamente 1/16 do tempo utilizado pelo POD-Clus e cerca de 1/19 do tempo gasto pelo eFCC.

Contudo, o tempo de processamento do ECM não engloba todo o tempo gasto com processamento, tendo em vista que o ECM lida apenas com fluxos de dados unidimensionais carecendo, desse modo, que seja gerado previamente um fluxo unidimensional, nesse caso de dimensões fractais, para cada um dos fluxos de dados e esses fornecidos como entrada para o ECM. Já o tempo elevado dispendido pelo POD-Clus, se registra devido ao fato de cada uma das dimensões dos fluxos de dados ser tratada independentemente uma das outras e então agrupadas

Figura 39 – Tempo gasto de processamento em minutos para cada um dos métodos para processar todo o conjunto de dados (21024000 medidas).

Fonte: Elaborada pelo autor.

aquelas que possuem similaridade em todas as dimensões. o eFCC, embora construa grupos mais coesos e com melhor distribuição dos dados que o eFCDS, o teste de expansão e contração, resulta em diversos cálculos da dimensão fractal, para decidir onde agrupar o fluxo de dados.

Por consequência, o tempo gasto pelo eFCDS para processar cada dado foi de aproximadamente  $10^{-6}$  segundos, isto é, o eFCDS é capaz de processar em torno de 100000 dados por segundo, permitindo, de fato, o processamento da fluxos de dados em tempo real. Já o eFCC, embora tenha apresentado melhores resultados que o eFCDS, contudo o tempo gasto de aproximadamente 1 minuto e 30 segundos para processar cada ano de dados, indica que o eFCC ainda pode sofrer melhorias nesse quesito.

Outro teste aplicado ao eFCC foi a capacidade de identificar e tratar fluxos de dados anômalos dentro dos grupos formados. Para isso foi criado um arquivo de registro de atividades. Esse arquivo anotou quando um fluxo de dados, pertencente a um agrupamento, era considerado anômalo e também registrou as estatísticas anterior e posterior a remoção do fluxo de dados. Nesse arquivo foi possível observar que em média, a cada iteração, 4% dos fluxos de dados tiveram que ser removidos de seus grupos. Esse fluxos de dados ou foram adicionado em outros grupos ou permaneceram como anômalos. De modo geral, a detecção de fluxos de dados anômalos interna aos grupos formados contribuiu para melhora da coesão dos grupos formados em cerca de 3,5%.

Os resultados apresentados nessa seção, foram analisados em parceria com especialistas da Embrapa-Campinas. Avaliações empíricas dos resultados indicaram que agrupar fluxos de dados (sensores climáticos) de modo dinâmico e acompanhando a evolução dos dados, auxiliam a encontrar regiões com clima homogêneo e também identificar mudanças que ocorrem ao longo do tempo. Essa informação pode ser útil para:

- Estudos de mudança climática - estudos dos fenômenos climáticos, tais como *El Niño* e *La Niña*, que usualmente alteram o padrão climático, regional e globalmente, e podem

impactar negativamente o clima, apresentando variações bruscas de temperatura ou no regime de chuvas.

- Contribuir com o plantio de culturas agrícolas - ao identificar as mudanças climáticas e as regiões homogêneas é possível planejar melhor o tipo de cultura agrícola a ser produzida.

## 6.3 Considerações finais

Como foi possível observar nesse capítulo, ao realizar o comparativo dos métodos propostos nesse doutoramento com os métodos propostos na literatura e que buscam realizar o agrupamento de fluxos dados, utilizando um conjunto sintético de fluxos de dados multidimensionais, tanto o eFCDS quanto o eFCC agrupam muito bem formas diferentes em grupos distintos, que seria a opção mais lógica mas que tanto o ECM quanto o PODClus não conseguem representar. Assim como o eFCDS, o eFCC também mostrou-se muito resistente em agrupar formas distintas em um mesmo grupo, ambos necessitando de valores elevados para o desvio padrão máximo e o eFCC de valores elevados também para o limiar de expansão e contração. Nesse conjunto de dados, o eFCDS atingiu 75% de precisão ao tentar representar as situações propostas e o eFCC conseguiu representar todas as situações, obtendo 100% de precisão.

Já os resultados atingidos pelo eFCDS e pelo eFCC, ao realizar o agrupamento de fluxos de dados reais, demonstram que ambos conseguem formar grupos com qualidade muito superior os concorrentes analisados e altamente coesos. O eFCC obtendo, inclusive, grupos 11% mais coesos que o eFCDS. Todavia o eFCDS requer um tempo de processamento muito menor que eFCC sendo indicado para processar fluxos de dados em tempo real. Já o eFCC é indicado para processar os fluxos de dados cuja frequência de geração dos dados não seja tão elevada, mas que a qualidade dos agrupamentos formados seja fundamental.

Através dos experimentos realizados foi possível observar também que ao utilizar a dimensão fractal como critério de formação dos agrupamentos e como critério para verificar a correlação entre as dimensões dos fluxos de dados, é possível realizar a construção de grupos mais coesos e de melhor qualidade do que se utilizar medidas descritivas básicas para tais tarefas. Desse modo, respondendo as questões de pesquisa 1 e 2, pode-se afirmar que é possível realizar a construção de agrupamentos de fluxos de dados multidimensionais por meio do uso da dimensão fractal para correlacionar as dimensões dos fluxos de dados e como critério de agrupamento para se obter grupos mais coesos. Além disso, é possível realizar o processamento de fluxos de dados multidimensionais em tempo real utilizando-se a dimensão fractal, respondendo, desse modo, a questão de pesquisa de número 3.





---

## CONCLUSÃO

---

Extrair conhecimento de fluxo de dados é de grande importância nos dias de hoje, pois a cada dia incontáveis novas medidas são geradas pelos mais variados tipos de sensores, tais como, monitores de atividade física, sensores de segurança ou sensores meteorológicos. Em geral, os fluxos de dados gerados por esses sensores são multidimensionais e sem um fim estabelecido. Uma tarefa comumente utilizada para extrair conhecimento desse tipo de dados é realizar o agrupamento de dados, como uma profusão de métodos desenvolvidos para esse fim. Contudo, poucos métodos foram desenvolvidos para realizar o agrupamento de fluxo de dados como um todo e apenas o trabalho apresentado por [Chaovalit \(2012\)](#) havia considerado fluxos de dados multidimensionais. Esses trabalhos apresentam diversas limitações, que constituíram a principal motivação para o desenvolvimento dessa tese de doutoramento.

As principais oportunidades de exploração e busca de solução para preenchimento das lacunas de pesquisa relacionadas a agrupamento de fluxos de dados foram:

1. Explorar alternativas as medidas descritivas comumente utilizadas para a construção dos agrupamentos.
2. Suportar o processamento de fluxos de dados multidimensionais.
3. Avaliar a correlação existente entre as diversas dimensões dos fluxos de dados para então construir os agrupamentos.
4. Acompanhar a característica evolutiva dos fluxos de dados.
5. Tratar períodos anômalos dos fluxos de dados.
6. Tratar fluxos de dados em tempo real.

Com relação ao primeiro item, a grande maioria dos métodos presentes na literatura utilizam medidas descritivas para representar o fluxo de dados e constroem seus agrupamentos

com base nessas medidas. Entretanto, o uso de medidas descritivas pode mascarar comportamentos anômalos que possam ocorrer em determinados períodos dos fluxos de dados, afetando a qualidade dos grupos gerados. Quanto ao segundo item, conforme apresentado, apenas um método realiza o agrupamento de fluxo de dados multidimensionais. Contudo, os agrupamentos gerados por esse método, em muitos casos, são de baixa qualidade, conforme apresentaram os experimentos realizados. Além disso, referente ao terceiro item, esse único método existente, até então, não correlaciona as dimensões dos fluxos de dados, sendo uma das causas para a baixa qualidade dos agrupamentos gerados por ele. O quarto e quinto itens são abordados pelos métodos existentes, todavia, são utilizadas soluções simplistas, como determinar um grupo de *outliers* apenas pela quantidade de membros existentes no grupo ou determinar a união ou divisão de um grupo apenas pela proximidade dos membros dos grupos. O sexto item foi considerado devido ao tempo de processamento gasto pelos métodos para processar um conjunto de fluxo de dados ser elevado, inviabilizando-os ao tratamento de fluxos de dados multidimensionais em tempo real.

Diante desse cenário, o presente trabalho de doutoramento buscou contribuir com dois novos métodos de agrupamento de fluxos de dados que lidassem com a multidimensionalidade e as correlações presentes nos dados, que representassem o comportamento dinâmico dos dados de modo incremental e que dessem o devido tratamento aos fluxos de dados que apresentassem algum desvio de comportamento dentro do período analisado. O primeiro, o método eFCDS, contribuiu com o agrupamento e processamento em tempo real de fluxos de dados multidimensionais, correlacionando as diversas dimensões existentes. O segundo método, o eFCC, considerou, também, a distribuição dos dados para realizar a construção dos agrupamentos e identificou *outliers* internos aos grupos gerados. A seguir, na Seção 7.1, são detalhadas as contribuições desta tese.

## 7.1 Principais Contribuições

Esta tese apresenta contribuições na área de mineração de fluxos de dados, propondo métodos de agrupamento de fluxos de dados multidimensionais que abordam soluções para as principais lacunas de pesquisa nesse tópico, em conformidade com o estado da arte.

Assim, neste trabalho de doutoramento foram desenvolvidos dois novos métodos para agrupamento de fluxos de dados. O primeiro, nomeado eFCDS (*Evolving Fractal Clustering of Data Streams*) adota o conceito da dimensão fractal de correlação ( $\mathcal{D}_2$ ) para mensurar a correlação, linear ou não, existente entre as dimensões de um fluxo de dados multidimensional, em um determinado período de tempo. Essa medida é utilizada como critério para agrupar fluxos de dados com comportamento semelhante ao longo do tempo. O segundo método, nomeado eFCC (*evolving Fractal Clusters Construction*) realiza o agrupamento de fluxos de dados multidimensionais de acordo com dois critérios principais: o comportamento semelhante ao

longo do tempo e distribuição de dados similar.

Os métodos foram comparados com outros métodos de agrupamento de fluxos de dados existentes no estado da arte, utilizando um conjunto de fluxos de dados sintético e outro real, com o objetivo de aferir comportamentos distintos utilizando diferentes tipos de dados. Tanto o eFCDS quanto o eFCC mostraram capacidade de agrupar fluxos de dados de 1,5 a 2 vezes melhor que os principais concorrentes da literatura. Eles obtiveram grupos mais coesos e de melhor qualidade.

Além disso, o eFCDS provou ser da ordem de três a vinte vezes mais rápido que seus concorrentes, habilitando-o a realizar o processamento de fluxos de dados em tempo real. Já o eFCC provou ser um método robusto, aplicável onde a qualidade dos agrupamentos gerados seja primordial, além de identificar fluxos de dados anômalos dentro dos agrupamentos gerados.

Ambos os métodos também mostrar ser ferramentas úteis no auxílio à pesquisadores da área de agrometeorologia que estudam o comportamento climático durante períodos de tempo, sem a necessidade de analisar todo o conjunto de dados manualmente.

Sumarizando, as contribuições provenientes dessa tese de doutoramento foram:

- O método eFCDS para construção de agrupamentos de fluxos de dados em tempo real.
- O método eFCC para construir agrupamentos com maior qualidade e coesão.
- Uma abordagem baseada em fractais para realizar o tratamento adequado de fluxos de dados multidimensionais, correlacionando as diversas dimensões.
- Método BRO para detecção de *outliers* interno aos grupos formados.
- Construção de um conjunto de fluxos de dados sintético, com diferentes formas para avaliação de métodos de agrupamento de fluxos de dados.
- Avaliação experimental extensiva dos métodos propostos, aplicando uma abrangente variação de parâmetros para identificar a influência de cada um na construção dos grupos. Onde mostrou-se que as questões de pesquisa puderam ser respondidas. Pois, através do uso da dimensão fractal foi possível realizar o processamento em tempo real de fluxos de dados multidimensionais e obteve-se melhores resultados de coesão dos grupos formados.
- Auxílio a especialistas de domínio, da Embrapa, na análise de fluxos de dados extensos.

## 7.2 Publicações

Durante o período de doutoramento, foram produzidos três artigos para divulgação dos resultados obtidos, sendo um artigo publicado em periódico nacional, um artigo publicado e

apresentado em conferência nacional e um artigo publicado e apresentado em conferência internacional. Além disso, está em elaboração um artigo completo para apresentação dos resultados finais do trabalho, a ser submetido a um periódico internacional. As publicações geradas até o momento são:

- BONES, C. C.; ROMANI, L. A. S. ; SOUSA, E. P. M. . Clustering Multivariate Data Streams by Correlating Attributes using Fractal Dimension. *Journal of Information and Data Management - JIDM*, v. 7, p. 249-264, 2016.
- BONES, C. C.; ROMANI, L. A. S. ; SOUSA, E. P. M. . Improving Multivariate Data Streams Clustering. In: *The International Conference on Computational Science (ICCS 2016)*, 2016, San Diego, CA, USA. *Procedia Computer Science*, 2016. v. 80. p. 461-471.
- BONES, C. C.; ROMANI, L. A. S. ; SOUSA, E. P. M. . Clustering Multivariate Climate Data Streams using Fractal Dimension. In: *XXX Simpósio Brasileiro de Banco de Dados*, 2015, Petrópolis. *Proceedings of the 30th Brazilian Symposium on Databases*, 2015. v. 30. p. 41-52.

### 7.3 Propostas para Trabalhos Futuros

As principais propostas para trabalhos futuros são sumarizadas a seguir.

- **Aplicação em outros domínios de aplicação:** Uma das primeiras derivações dessa tese de doutoramento é a aplicação dos métodos desenvolvidos em fluxo de dados da área de saúde, por exemplo, fluxos de dados derivados de sensores de atividades físicas.
- **Aperfeiçoamento da complexidade computacional:** Habilitar o método eFCC para realizar o agrupamento de fluxos de dados em tempo real.
- **Tratar fluxos de dados com dimensionalidades distintas:** Ao tratar fluxos de dados com quantidades diferentes de dimensões habilitará os métodos a serem mais tolerantes a falhas, quando alguma das medidas geradas pelos sensores falhar.

## REFERÊNCIAS

---

ACKERMANN, M. R.; MÄRTENS, M.; RAUPACH, C.; SWIERKOT, K.; LAMMERSEN, C.; SOHLER, C. Streamkm++: A clustering algorithm for data streams. **ACM Journal of Experimental Algorithmics**, ACM, New York, NY, USA, v. 17, p. 1–30, maio 2012. Citado nas páginas [34](#), [40](#), [54](#) e [58](#).

AGGARWAL, C. C. (Ed.). **Data Streams - Models and Algorithms**. [S.l.]: Springer, 2007. v. 31. (Advances in Database Systems, v. 31). ISBN 978-0-387-28759-1. Citado nas páginas [33](#) e [71](#).

AGGARWAL, C. C. A segment-based framework for modeling and mining data streams. **Knowledge and Information Systems**, Springer, v. 30, n. 1, p. 1–29, jan. 2012. Citado na página [45](#).

\_\_\_\_\_. Mining sensor data streams. In: AGGARWAL, C. C. (Ed.). **Managing and Mining Sensor Data**. New York: Springer Science & Business Media, 2013. cap. 6, p. 143–171. Citado nas páginas [31](#), [33](#) e [41](#).

AGGARWAL, C. C.; HAN, J.; WANG, J.; YU, P. S. A framework for clustering evolving data streams. In: **Proceedings of the 29th International Conference on Very Large Data Bases**. Berlin, Germany: VLDB Endowment, 2003. (VLDB '03, v. 29), p. 81–92. Citado nas páginas [34](#), [54](#), [58](#) e [59](#).

\_\_\_\_\_. On demand classification of data streams. In: **Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. New York, NY, USA: ACM, 2004. (KDD '04), p. 503–508. Citado na página [54](#).

\_\_\_\_\_. A framework for on-demand classification of evolving data streams. **IEEE Transactions on Knowledge and Data Engineering**, v. 18, n. 5, p. 577–589, maio 2006. Citado nas páginas [31](#), [32](#) e [54](#).

\_\_\_\_\_. On clustering massive data streams: Summarization paradigm. In: AGGARWAL, C. C. (Ed.). **Data Streams - Models and Algorithms**. Yorktown Heights, NY, USA: Springer, 2007, (Advances in Database Systems, v. 31). cap. 2, p. 169–208. ISBN 978-0-387-28759-1. Citado nas páginas [40](#) e [53](#).

AGGARWAL, C. C.; YU, P. S. A survey of synopsis construction in data streams. In: AGGARWAL, C. C. (Ed.). **Data Streams - Models and Algorithms**. Yorktown Heights, NY, USA: Springer, 2007, (Advances in Database Systems, v. 31). cap. 9, p. 169–208. ISBN 978-0-387-28759-1. Citado nas páginas [42](#) e [47](#).

AI, T.; ZHANG, R.; ZHOU, H.; PEI, J. Box-counting methods to directly estimate the fractal dimension of a rock surface. **Applied Surface Science**, Elsevier, v. 314, p. 610–621, 2014. Citado na página [77](#).

- AMIGÓ, E.; GONZALO, J.; ARTILES, J.; VERDEJO, F. A comparison of extrinsic clustering evaluation metrics based on formal constraints. **Information Retrieval**, Springer, v. 12, n. 4, p. 461–486, 2009. Citado nas páginas 71 e 74.
- AMINI, A.; WAH, T. Y.; SABOOHI, H. On density-based data streams clustering algorithms: a survey. **Journal of Computer Science and Technology**, Springer, v. 29, n. 1, p. 116–141, 2014. Citado nas páginas 58, 60 e 74.
- ARANGANAYAGI, S.; THANGAVEL, K. Clustering categorical data using silhouette coefficient as a relocating measure. In: **International Conference on Computational Intelligence and Multimedia Applications**. Sivakasi, Tamilnadu, India: IEEE, 2007. v. 2, p. 13–17. Citado na página 40.
- ARBELAITZ, O.; GURRUTXAGA, I.; MUGUERZA, J.; PÉREZ, J. M.; PERONA, I. An extensive comparative study of cluster validity indices. **Pattern Recognition**, Elsevier, v. 46, n. 1, p. 243–256, 2013. Citado nas páginas 71, 73 e 74.
- AZEMIN, M. Z. C.; HAMID, F. A.; WANG, J. J.; KAWASAKI, R.; KUMAR, D. K. Box-counting fractal dimension algorithm variations on retina images. In: **Advanced Computer and Communication Engineering Technology**. [S.l.]: Springer, 2016. p. 337–343. Citado na página 77.
- BABCOCK, B.; BABU, S.; DATAR, M.; MOTWANI, R.; WIDOM, J. Models and issues in data stream systems. In: **Proceedings of the 21st ACM SIGMOD-SIGACT-SIGART symposium on Principles of Database Systems**. Madison, Wisconsin, USA: ACM, 2002. p. 1–16. Citado nas páginas 41 e 53.
- BARBARA, D. Requirements for clustering data streams. **ACM SIGKDD Explorations Newsletter**, ACM, New York, NY, USA, v. 3, n. 2, p. 23–27, jan. 2002. Citado nas páginas 53, 54 e 55.
- BARBARA, D.; CHEN, P. Fractal mining-self similarity-based clustering and its applications. In: **Data Mining and Knowledge Discovery Handbook**. [S.l.]: Springer, 2009. p. 573–589. Citado nas páginas 35, 36 e 61.
- BELUSSI, A.; FALOUTSOS, C. Estimating the selectivity of spatial queries using the correlation' fractal dimension. In: **The 21th International Conference on Very Large Data Bases**. Zurich, Switzerland.: [s.n.], 1995. v. 21, p. 299–310. Citado nas páginas 79, 80 e 81.
- BERINGER, J.; HÜLLERMEIER, E. Online clustering of parallel data streams. **Data & Knowledge Engineering**, Elsevier, v. 58, n. 2, p. 180–204, 2006. ISSN 0169-023X. Citado nas páginas 48, 49, 54, 58 e 63.
- BIFET, A.; GAVALDA, R. Learning from time-changing data with adaptive windowing. In: **Proceedings of the 2007 SIAM International Conference on Data Mining**. Minneapolis, Minnesota, USA: SIAM, 2007. v. 7, p. 443–448. Citado nas páginas 42 e 43.
- BIRANT, D.; KUT, A. St-dbscan: An algorithm for clustering spatial-temporal data. **Data & Knowledge Engineering**, Elsevier, v. 60, n. 1, p. 208–221, 2007. Citado nas páginas 40 e 58.
- BONES C. C.; ROMANI, L. A. S. S. E. P. M. Clustering multivariate climate data streams using fractal dimension. In: **SBC. Proceedings of the 30th Brazilian Symposium on Databases**. Petrópolis - RJ - Brazil, 2015. v. 30, p. 41–52. Citado na página 36.

\_\_\_\_\_. Clustering multivariate data streams by correlating attributes using fractal dimension. **Journal of Information and Data Management - JIDM**, v. 7, n. 3, p. 249–264, 2016. Citado na página 36.

\_\_\_\_\_. Improving multivariate data streams clustering. In: **Procedia Computer Science**. San Diego, CA, USA: [s.n.], 2016. (The International Conference on Computational Science (2016), v. 80), p. 461–471. Citado na página 36.

BRAVERMAN, V.; OSTROVSKY, R.; ZANIOLO, C. Optimal sampling from sliding windows. **Journal of Computer and System Sciences**, Elsevier, v. 78, n. 1, p. 260–272, 2012. Citado na página 42.

BRUN, M.; SIMA, C.; HUA, J.; LOWEY, J.; CARROLL, B.; SUH, E.; DOUGHERTY, E. R. Model-based evaluation of clustering validation measures. **Pattern Recognition**, v. 40, n. 3, p. 807 – 824, 2007. ISSN 0031-3203. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0031320306003104>>. Citado nas páginas 71, 72, 73 e 74.

CAO, F.; ESTER, M.; QIAN, W.; ZHOU, A. Density-based clustering over an evolving data stream with noise. In: **Proceedings of the 2006 SIAM International Conference on Data Mining**. Bethesda, Maryland, USA: SIAM, 2006. v. 6, p. 326–337. Citado nas páginas 48, 49 e 58.

CHAIRUKWATTANA, R.; KANGKACHIT, T.; RAKTHANMANON, T.; WAIYAMAI, K. Se-stream: Dimension projection for evolution-based clustering of high dimensional data streams. In: HUYNH, V. N.; DENOEU, T.; TRAN, D. H.; LE, A. C.; PHAM, S. B. (Ed.). **Knowledge and Systems Engineering**. [S.l.]: Springer, 2014, (Advances in Intelligent Systems and Computing, v. 245). p. 365–376. Citado nas páginas 40 e 54.

CHAKRABARTI, D.; KUMAR, R.; TOMKINS, A. Evolutionary clustering. In: **Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. Philadelphia, PA, USA: ACM, 2006. p. 554–560. ISBN 1-59593-339-5. Citado na página 58.

CHAOVALIT, P. **Clustering transient data streams by example and by variable**. Tese (Tese de Doutorado) — University of Maryland, 2009. Citado nas páginas 54, 55 e 71.

CHAOVALIT, P. An effective streams clustering method for biomedical signals. In: **Innovations in Data Methodologies and Computational Algorithms for Medical Applications**. [S.l.]: IGI Global, 2012. p. 153–183. Citado nas páginas 33, 34, 41, 56, 63, 67, 68, 71, 72, 89, 108 e 127.

CHARIKAR, M.; CHEN, K.; FARACH-COLTON, M. Finding frequent items in data streams. **Automata, Languages and Programming**, Springer, v. 2380, p. 693–703, 2002. Citado na página 54.

CI, S.; GUIZANI, M.; SHARIF, H. Adaptive clustering in wireless sensor networks by mining sensor energy data. **Computer Communications**, Elsevier, Amsterdam, Holanda, v. 30, n. 14-15, p. 2968–2975, out. 2007. Citado na página 40.

CORMODE, G.; GAROFALAKIS, M.; HAAS, P. J.; JERMAINE, C. Synopses for massive data: Samples, histograms, wavelets, sketches. **Foundations and Trends in Databases**, Now Publishers Inc, v. 4, n. 1–3, p. 1–294, 2012. Citado nas páginas 44, 45, 46, 47, 48, 89 e 97.



- CORMODE, G.; HADJIELEFTHERIOU, M. Methods for finding frequent items in data streams. **The International Journal on Very Large Data Bases**, Springer, v. 19, n. 1, p. 3–20, fev. 2010. Citado nas páginas 44 e 54.
- DASGUPTA, S.; LONG, P. M. Performance guarantees for hierarchical clustering. **Journal of Computer and System Sciences**, Elsevier, v. 70, n. 4, p. 555–569, 2005. Citado na página 57.
- DATAR, M.; GIONIS, A.; INDYK, P.; MOTWANI, R. Maintaining stream statistics over sliding windows. **Journal on Computing**, SIAM, v. 31, n. 6, p. 1794–1813, 2002. Citado na página 42.
- DING, H.; TRAJCEVSKI, G.; SCHEUERMANN, P.; WANG, X.; KEOGH, E. Querying and mining of time series data: Experimental comparison of representations and distance measures. **The International Journal on Very Large Data Bases**, VLDB Endowment, v. 1, n. 2, p. 1542–1552, ago. 2008. ISSN 2150-8097. Citado na página 49.
- DING, S.; ZHANG, J.; JIA, H.; QIAN, J. An adaptive density data stream clustering algorithm. **Cognitive Computation**, v. 8, n. 1, p. 30–38, Feb 2016. ISSN 1866-9964. Disponível em: <<https://doi.org/10.1007/s12559-015-9342-z>>. Citado nas páginas 29, 30 e 54.
- ELMASRI, R.; NAVATHE, S. **Sistemas de Banco de Dados**. São Paulo, SP: Person, 2011. v. 6. 1200 p. Citado na página 41.
- ERIKSSON, B.; DASARATHY, G.; SINGH, A.; NOWAK, R. Active clustering: Robust and efficient hierarchical clustering using adaptively selected similarities. In: GORDON, D. D. G.; DUDAK, M. (Ed.). **14th International Conference on Artificial Intelligence and Statistics**. Ft. Lauderdale, Florida, USA: Microtome Publishing, 2011. v. 15, p. 260–268. Citado nas páginas 40, 54 e 58.
- FALOUTSOS, C. Data mining using fractals and power laws. In: DONG, G.; LIN, X.; WANG, W.; YANG, Y.; YU, J. X. (Ed.). **Advances in Data and Web Management**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007. p. 1–1. ISBN 978-3-540-72524-4. Citado na página 35.
- FALOUTSOS, C.; KAMEL, I. Beyond uniformity and independence: Analysis of r-trees using the concept of fractal dimension. In: **Proceedings of the Thirteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems**. Minneapolis, Minnesota, USA: ACM, 1994. p. 4–13. Citado nas páginas 35 e 77.
- FAYYAD, U.; PIATETSKY-SHAPIO, G.; SMYTH, P. From data mining to knowledge discovery in databases. **AI magazine**, v. 17, n. 3, p. 37, 1996. Citado nas páginas 15, 32, 53 e 54.
- FELDMAN, D.; LANGBERG, M. A unified framework for approximating and clustering data. In: **43rd annual ACM symposium on Theory of computing**. San Jose, California, USA: ACM, 2011. p. 569–578. Citado nas páginas 54 e 58.
- FORESTIERO, A.; PIZZUTI, C.; SPEZZANO, G. A single pass algorithm for clustering evolving data streams based on swarm intelligence. **Data Mining and Knowledge Discovery**, Springer, v. 26, n. 1, p. 1–26, 2013. Citado nas páginas 29, 30, 40, 54 e 58.
- GABER, M. M.; ZASLAVSKY, A.; KRISHNASWAMY, S. Mining data streams: a review. **ACM Sigmod Record**, ACM, v. 34, n. 2, p. 18–26, 2005. Citado nas páginas 42, 54 e 58.



\_\_\_\_\_. A survey of classification methods in data streams. In: AGGARWAL, C. C. (Ed.). **Data Streams - Models and Algorithms**. Yorktown Heights, NY, USA: Springer, 2007, (Advances in Database Systems, v. 31). cap. 3, p. 39–60. ISBN 978-0-387-28759-1. Citado na página 54.

GAMA, J. **Knowledge Discovery from Data Streams**. Boca Raton, Florida, USA: Chapman & Hall Book, 2010. 233 p. Citado nas páginas 29, 31, 32, 33, 39, 41, 42, 43, 44, 45, 46, 47, 51, 53, 55, 56, 58, 62, 64 e 71.

\_\_\_\_\_. A survey on learning from data streams: current and future trends. **Progress in Artificial Intelligence**, Springer, v. 1, n. 1, p. 45–55, 2012. Citado nas páginas 29, 53 e 54.

GAMA, J.; GANGULY, A. R.; OMITAOMU, O. A.; VATSAVAI, R. R.; GABER, M. M. Knowledge discovery from data streams. **Journal Intelligent Data Analysis**, IOS Press, v. 13, n. 3, p. 403–404, 2009. Citado na página 54.

GAMA, J.; SEBASTIÃO, R.; RODRIGUES, P. P. On evaluating stream learning algorithms. **Machine Learning**, Springer, v. 90, n. 3, p. 317–346, 2013. Citado na página 56.

GANGULY, A. R.; STEINHAEUSER, K. Data mining for climate change and impacts. In: **International Conference on Data Mining Workshops 2008**. Pisa, Italia: IEEE, 2008. p. 385–394. Citado na página 33.

GILBERT, A. C.; KOTIDIS, Y.; MUTHUKRISHNAN, S.; STRAUSS, M. One-pass wavelet decompositions of data streams. **Transactions on Knowledge and Data Engineering**, IEEE, v. 15, n. 3, p. 541–554, 2003. Citado nas páginas 45 e 46.

GOLAB, L. **Sliding Window Query Processing over Data Streams**. Tese (Doutorado) — University of Waterloo, 2006. Disponível em: <<https://uwspace.uwaterloo.ca/bitstream/10012/2930/1/lgolab2006.pdf>>. Acesso em: 30.7.2014. Citado na página 43.

GOLAB, L. Data stream. In: LIU, L.; ÖZSU, M. T. (Ed.). **Encyclopedia of Database Systems**. New York, NY, USA: Springer-Verlag, 2009. p. 638. Citado nas páginas 30, 39, 41 e 53.

GUHA, S. Tight results for clustering and summarizing data streams. In: **Proceedings of the 12th International Conference on Database Theory**. New York, NY, USA: ACM, 2009. (ICDT '09), p. 268–275. Citado nas páginas 40, 45 e 58.

GUHA, S.; KOUDAS, N.; SHIM, K. Approximation and streaming algorithms for histogram construction problems. **ACM Trans. Database Syst.**, ACM, New York, NY, USA, v. 31, n. 1, p. 396–438, mar. 2006. Citado na página 45.

GUHA, S.; MEYERSON, A.; MISHRA, N.; MOTWANI, R.; O'CALLAGHAN, L. Clustering data streams: Theory and practice. **Transactions on Knowledge and Data Engineering**, IEEE, v. 15, n. 3, p. 515–528, 2003. Citado nas páginas 30, 34, 39, 41, 55, 56, 58, 59 e 60.

GUHA, S.; MISHRA, N.; MOTWANI, R.; O'CALLAGHAN, L. Clustering data streams. In: **Proceedings 41st annual Symposium on Foundations of Computer Science**. Redondo Beach, California, USA: IEEE, 2000. p. 359–366. Citado na página 58.

GUHA, S.; SHIM, K.; WOO, J. Rehist: Relative error histogram construction algorithms. In: **Proceedings of the 30th international conference on Very Large Data Bases**. Toronto, Ontario, Canada: Morgan Kaufmann Publishers, 2004. v. 30, p. 300–311. Citado na página 45.

- GUO, H. Exploring online learning data using fractal dimensions. **ETS Research Report Series**, Wiley Online Library, 2017. Citado na página 77.
- HAHSLER, M.; DUNHAM, M. H. Temporal structure learning for clustering massive data streams in real-time. In: **Proceedings of the 2011 SIAM International Conference on Data Mining**. Mesa, Arizona, USA: SIAM, 2011. p. 664–675. Citado nas páginas 54, 56 e 58.
- HAN, J.; PEI, J.; KAMBER, M. **Data mining: concepts and techniques**. 3. ed. Boston, MA, USA: Elsevier, 2011. Citado nas páginas 60 e 71.
- HOEFFDING, W. Probability inequalities for sums of bounded random variables. **Journal of the American Statistical Association**, Taylor & Francis Group, v. 58, n. 301, p. 13–30, 1963. Citado na página 65.
- HRUSCHKA, E. R.; CAMPELLO, R. J.; CASTRO, L. N. de. Improving the efficiency of a clustering genetic algorithm. In: SPRINGER. **Ibero-American Conference on Artificial Intelligence**. [S.l.], 2004. p. 861–870. Citado nas páginas 73 e 74.
- HRUSCHKA, E. R.; CAMPELLO, R. J.; FREITAS, A. A. *et al.* A survey of evolutionary algorithms for clustering. **IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)**, IEEE, v. 39, n. 2, p. 133–155, 2009. Citado na página 74.
- HYDE, R.; ANGELOV, P.; MACKENZIE, A. Fully online clustering of evolving data streams into arbitrarily shaped clusters. **Information Sciences**, v. 382-383, n. Supplement C, p. 96 – 114, 2017. ISSN 0020-0255. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0020025516319247>>. Citado nas páginas 29, 30, 34, 40 e 54.
- JAIN, A. K. Data clustering: 50 years beyond k-means. **Pattern Recognition Letters**, Elsevier, v. 31, n. 8, p. 651–666, 2010. Citado na página 58.
- JAWERTH, B.; SWELDENS, W. An overview of wavelet based multiresolution analyses. **SIAM review**, SIAM, v. 36, n. 3, p. 377–412, 1994. Citado nas páginas 45, 47 e 48.
- KAMEL, I.; FALOUTSOS, C. Hilbert r-tree: An improved r-tree using fractals. In: **Proceedings of the 20th International Conference on Very Large Data Bases**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1994. (VLDB '94), p. 500–509. Citado nas páginas 35 e 36.
- KEOGH, E.; WEI, L.; XI, X.; LEE, S.-H.; VLACHOS, M. Lb\_keogh supports exact indexing of shapes under rotation invariance with arbitrary representations and distance measures. In: VLDB ENDOWMENT. **Proceedings of the 32nd International Conference on Very Large Data Bases**. Seoul, Korea: ACM, 2006. p. 882–893. Citado na página 51.
- KHAN, I.; HUANG, J. Z.; IVANOV, K. Incremental density-based ensemble clustering over evolving data streams. **Neurocomputing**, v. 191, n. Supplement C, p. 34 – 43, 2016. ISSN 0925-2312. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0925231216000564>>. Citado nas páginas 29, 30, 34 e 40.
- KRANEN, P.; ASSENT, I.; BALDAUF, C.; SEIDL, T. The clustree: indexing micro-clusters for anytime stream mining. **Knowledge and Information Systems**, v. 29, n. 2, p. 249–272, Nov 2011. ISSN 0219-3116. Disponível em: <<https://doi.org/10.1007/s10115-010-0342-8>>. Citado na página 34.

- KREMER, H.; KRANEN, P.; JANSEN, T.; SEIDL, T.; BIFET, A.; HOLMES, G.; PFAHRINGER, B. An effective evaluation measure for clustering on evolving data streams. In: **Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. San Diego, California, USA: ACM, 2011. p. 868–876. Citado na página 40.
- KRISHNAMURTHY, A.; BALAKRISHNAN, S.; XU, M.; SINGH, A. Efficient active algorithms for hierarchical clustering. In: **Proceedings of the 29th International Conference on Machine Learning**. Edinburgh, Scotland, UK: Omnipress, 2012. p. 8. Citado nas páginas 40, 54 e 57.
- LEE, P.; LAKSHMANAN, L.; MILIOS, E. Incremental cluster evolution tracking from highly dynamic network data. In: **30th International Conference on Data Engineering**. Chicago, Illinois, USA: IEEE, 2014. p. 3–14. Citado na página 40.
- LI, H.-F.; LEE, S.-Y. Mining frequent itemsets over data streams using efficient window sliding techniques. **Expert Systems with Applications**, Elsevier, v. 36, n. 2, Part 1, p. 1466–1477, 2009. Citado nas páginas 42 e 54.
- LIBEN-NOWELL, D.; VEE, E.; ZHU, A. Finding longest increasing and common subsequences in streaming data. **Journal of Combinatorial Optimization**, Springer, v. 11, n. 2, p. 155–175, 2006. Citado nas páginas 48 e 49.
- LIU, B. Database redundant attribute detection using fractal dimension. In: **2014 IEEE 5th International Conference on Software Engineering and Service Science**. [S.l.: s.n.], 2014. p. 561–564. Citado nas páginas 35 e 36.
- LIU, Y.; LI, Z.; XIONG, H.; GAO, X.; WU, J. Understanding of internal clustering validation measures. In: IEEE. **Data Mining (ICDM), 2010 IEEE 10th International Conference on**. [S.l.], 2010. p. 911–916. Citado nas páginas 71, 73 e 74.
- MANDELBROT, B. B. The fractal geometry of nature. **Free**, 1982. Citado nas páginas 77, 78 e 79.
- MILLER, Z.; DICKINSON, B.; DEITRICK, W.; HU, W.; WANG, A. H. Twitter spammer detection using data stream clustering. **Information Sciences**, v. 260, p. 64–73, 2014. Citado nas páginas 40 e 58.
- NIRJON, S.; GREENWOOD, C.; TORRES, C.; ZHOU, S.; STANKOVIC, J. A.; YOON, H. J.; RA, H.-K.; BASARAN, C.; PARK, T.; SON, S. H. Kintense: A robust, accurate, real-time and evolving system for detecting aggressive actions from streaming 3d skeleton data. In: **IEEE International Conference on Pervasive Computing and Communications**. Budapest, Hungria: IEEE, 2014. p. 2–10. Citado na página 48.
- NUNES, S. A.; ROMANI, L. A.; AVILA, A. M.; COLTRI, P. P.; JR, C. T.; CORDEIRO, R. L.; SOUSA, E. P. de; TRAINA, A. J. Analysis of large scale climate data: how well climate change models and data from real sensor networks agree? In: **Proceedings of the 22nd International Conference on World Wide Web Companion**. Rio de Janeiro, Brazi: International World Wide Web Conferences Steering Committee, 2013. p. 517–526. Citado na página 42.
- PEREIRA, C. M. M. **Agrupamento de series temporais em fluxos continuos de dados**. Tese (Doutorado) — Universidade de Sao Paulo, 2013. Citado nas páginas 64, 68, 69 e 70.
- PEREIRA, C. M. M.; MELLO, R. F. de. Ts-stream: clustering time series on data streams. **Journal of Intelligent Information Systems**, Springer, v. 42, n. 3, p. 531–566, 2014. Citado nas páginas 29, 30, 34, 40, 63, 67 e 70.

- PUSCHMANN, D.; BARNAGHI, P.; TAFAZOLLI, R. Adaptive clustering for dynamic iot data streams. **IEEE Internet of Things Journal**, v. 4, n. 1, p. 64–74, Feb 2017. ISSN 2327-4662. Citado na página 40.
- RAKTHANMANON, T.; CAMPANA, B. J. L.; MUEEN, A.; BATISTA, G. E. A. P. A.; WESTOVER, M. B.; ZHU, Q.; ZAKARIA, J.; KEOGH, E. J. Searching and mining trillions of time series subsequences under dynamic time warping. In: **Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. Beijing, China: ACM, 2012. p. 262–270. Citado nas páginas 50 e 51.
- RATANAMAHATANA, C. A.; KEOGH, E. J. Making time-series classification more accurate using learned constraints. In: **Proceedings of the 2004 SIAM International Conference on Data Mining**. Lake Buena Vista, Florida, USA: SIAM, 2004. p. 11–22. Citado na página 51.
- REHMAN, M. Z. ur; LI, T.; YANG, Y.; WANG, H. Hyper-ellipsoidal clustering technique for evolving data stream. **Knowledge-Based Systems**, v. 70, n. Supplement C, p. 3 – 14, 2014. ISSN 0950-7051. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S095070511300381X>>. Citado nas páginas 34 e 40.
- RODRIGUES, P.; GAMA, J.; PEDROSO, J. Hierarchical clustering of time-series data streams. **Transactions on Knowledge and Data Engineering**, IEEE, v. 20, n. 5, p. 615–627, 2008. Citado nas páginas 34, 40, 63, 64, 66, 68, 69 e 71.
- RODRIGUES, P. P.; GAMA, J. Distributed clustering of ubiquitous data streams. **Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery**, Wiley Online Library, v. 4, n. 1, p. 38–54, 2014. Citado nas páginas 33 e 40.
- RODRIGUES, P. P.; GAMA, J.; LOPES, L. Clustering distributed sensor data streams. In: DAELEMANS, W.; MORIK, K. (Ed.). **Machine Learning and Knowledge Discovery in Databases**. New York, NY, USA: Springer, 2008. p. 282–297. Citado na página 40.
- RODRIGUES, P. P.; GAMA, J.; PEDROSO, J. P. Odac: Hierarchical clustering of time series data streams. In: SIAM. **Proceedings of the 2006 SIAM International Conference on Data Mining**. Bathesda, Maryland, USA, 2006. p. 499–503. Citado nas páginas 57 e 63.
- SCHROEDER, M. R. **Fractals, chaos, power laws: minutes from an infinite paradise**. New York, NY, USA: WH Freeman and Company, 1991. Citado nas páginas 35, 77, 78 e 81.
- SHEIKHOLESAMI, G.; CHATTERJEE, S.; ZHANG, A. Wavecluster: a wavelet-based clustering approach for spatial data in very large databases. **The VLDB Journal—The International Journal on Very Large Data Bases**, Springer-Verlag New York, Inc., v. 8, n. 3-4, p. 289–304, 2000. Citado na página 60.
- SHOUP, V. Lower bounds for discrete logarithms and related problems. In: **Advances in Cryptology**. Konstanz, Germany: Springer, 1997. p. 256–266. Citado na página 51.
- SILVA, A. D.; CHIKY, R.; HÉBRAIL, G. A clustering approach for sampling data streams in sensor networks. **Knowledge and information systems**, Springer, v. 32, n. 1, p. 1 – 23, 2012. Citado na página 40.
- SILVA, J. de A.; HRUSCHKA, E. R.; GAMA, J. An evolutionary algorithm for clustering data streams with a variable number of clusters. **Expert Systems with Applications**, v. 67, n. Supplement C, p. 228 – 238, 2017. ISSN 0957-4174. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0957417416304985>>. Citado nas páginas 29, 30, 34, 40 e 54.

- SONG, Q.; KASABOV, N. Ecm-a novel on-line, evolving clustering method and its applications. In: **Proceedings of the 5th Biannual Conference on Artificial Neural Networks and Expert Systems**. Dunedin, New Zealand: The MIT Press, 2001. p. 631–682. Citado na página 68.
- SOUSA, E. P. de; TRAINA, A. J.; JR, C. T.; FALOUTSOS, C. Measuring evolving data streams behavior through their intrinsic dimension. **New Generation Computing**, Springer, v. 25, n. 1, p. 33–60, 2007. Citado nas páginas 36, 77 e 83.
- SOUSA, E. P. M.; TRAINA, C.; TRAINA, A. J. M.; WU, L.; FALOUTSOS, C. A fast and effective method to find correlations among attributes in databases. **Data Mining and Knowledge Discovery**, v. 14, n. 3, p. 367–407, 2007. ISSN 1573-756X. Disponível em: <<http://dx.doi.org/10.1007/s10618-006-0056-4>>. Citado nas páginas 35, 36, 77, 79, 80, 81 e 85.
- SU, M.-C.; CHOU, C.-H. A modified version of the k-means algorithm with a distance based on cluster symmetry. **Transactions on Pattern Analysis and Machine Intelligence**, IEEE, v. 23, n. 6, p. 674–680, 2001. Citado na página 58.
- TOYODA, M.; SAKURAI, Y.; ISHIKAWA, Y. Pattern discovery in data streams under the time warping distance. **The International Journal on Very Large Data Bases**, Springer, v. 5, p. 1–24, 2013. Citado nas páginas 48, 49 e 54.
- TRAINA JR., C.; TRAINA, A. J. M.; WU, L.; FALOUTSOS, C. Fast feature selection using fractal dimension. In: **Proceedings of 15th Brazilian Symposium on Databases**. [S.l.: s.n.], 2000. p. 158–171. Citado na página 81.
- TRAN, D.-H.; SATTTLER, K.-U. Communication-efficient exact clustering of distributed streaming data. In: MURGANTE, B.; MISRA, S.; CARLINI, M.; TORRE, C.; NGUYEN, H.-Q.; TANIAR, D.; APDUHAN, B.; GERVASI, O. (Ed.). **Computational Science and Its Applications ICCSA 2013**. [S.l.]: Springer Berlin Heidelberg, 2013, (Lecture Notes in Computer Science, v. 7975). p. 421–436. Citado na página 58.
- TU, L.; CHEN, Y. Stream data clustering based on grid density and attraction. **ACM Trans. Knowl. Discov. Data**, ACM, New York, NY, USA, v. 3, n. 3, p. 12:1–12:27, jul. 2009. ISSN 1556-4681. Citado na página 60.
- TU, Q.; LU, J.; YUAN, B.; TANG, J.; YANG, J. Density-based hierarchical clustering for streaming data. **Pattern Recognition Letters**, Elsevier, v. 33, n. 5, p. 641–645, 2012. Citado nas páginas 40 e 58.
- WANG, C.-D.; LAI, J.-H.; HUANG, D.; ZHENG, W.-S. Svstream: A support vector-based algorithm for clustering data streams. **Transactions on Knowledge and Data Engineering**, IEEE, v. 25, n. 6, p. 1410–1424, June 2013. Citado nas páginas 40, 54 e 58.
- WANG, Q.; KORKIN, D.; SHANG, Y. A fast multiple longest common subsequence (mlcs) algorithm. **Transactions on Knowledge and Data Engineering**, IEEE, v. 23, n. 3, p. 321–334, March 2011. Citado nas páginas 48 e 49.
- WANG, W.; YANG, J.; MUNTZ, R. *et al.* Sting: A statistical information grid approach to spatial data mining. In: **VLDB**. [S.l.: s.n.], 1997. v. 97, p. 186–195. Citado na página 60.
- WIDIPUTRA, H.; PEARS, R.; KASABOV, N. Multiple time-series prediction through multiple time-series relationships profiling and clustered recurring trends. In: HUANG, J.; CAO, L.; SRIVASTAVA, J. (Ed.). **Advances in Knowledge Discovery and Data Mining**. Shenzhen, China: Springer, 2011. p. 161–172. Citado nas páginas 34, 40, 63, 68 e 108.



- WU, H.; SALZBERG, B.; SHARP, G. C.; JIANG, S. B.; SHIRATO, H.; KAEI, D. Subsequence matching on structured time series data. In: **Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data**. Baltimore, Maryland, USA: ACM, 2005. p. 682–693. Citado na página 49.
- WU, H.; SALZBERG, B.; ZHANG, D. Online event-driven subsequence matching over financial data streams. In: **Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data**. Paris, France: ACM, 2004. p. 23–34. Citado na página 49.
- YIN, J.; GABER, M. M. Clustering distributed time series in sensor networks. In: **International Conference on Data Mining**. Pisa, Italy: IEEE, 2008. p. 678–687. Citado na página 40.
- ZHANG, R. R. T.; LIVNY, M. Birch: an efficient data clustering method for very large databases. In: **ACM SIGMOD international Conference on Management of Data**. Montreal, Canada: ACM, 1996. v. 1, p. 103–114. Citado nas páginas 48, 49, 57, 58 e 59.
- ZHANG, X.; FURTLEHNER, C.; GERMAIN-RENAUD, C.; SEBAG, M. Data stream clustering with affinity propagation. **IEEE Transactions on Knowledge and Data Engineering**, v. 26, n. 7, p. 1644–1656, July 2014. ISSN 1041-4347. Citado na página 40.
- ZHANG, Y. D.; CHEN, X. Q.; ZHAN, T. M.; JIAO, Z. Q.; SUN, Y.; CHEN, Z. M.; YAO, Y.; FANG, L. T.; LV, Y. D.; WANG, S. H. Fractal dimension estimation for developing pathological brain detection system based on minkowski-bouligand method. **IEEE Access**, v. 4, p. 5937–5947, 2016. Citado na página 77.
- ZHENG, L.; HUO, H.; GUO, Y.; FANG, T. Supervised adaptive incremental clustering for data stream of chunks. **Neurocomputing**, v. 219, n. Supplement C, p. 502 – 517, 2017. ISSN 0925-2312. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S092523121631089X>>. Citado nas páginas 34 e 40.
- ZHOU, A.; CAO, F.; QIAN, W.; JIN, C. Tracking clusters in evolving data streams over sliding windows. **Knowledge and Information Systems**, Springer, v. 15, n. 2, p. 181–214, 2008. Citado na página 42.
- ZHU, L.; CAO, L.; YANG, J.; LEI, J. Evolving soft subspace clustering. **Applied Soft Computing**, v. 14, Part B, n. 0, p. 210–228, 2014. Evolving Soft Computing Techniques and Applications. Citado nas páginas 40, 54 e 58.