**Classificação de séries temporais utilizando diferentes
representações de dados e ensembles**

Rafael Giusti

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: _____

Rafael Giusti

Classificação de séries temporais utilizando diferentes representações de dados e ensembles

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

Orientador: Prof. Dr. Gustavo Enrique de Almeida Prado Alves Batista

USP – São Carlos
Outubro de 2017

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

G538c Giusti, Rafael
Classificação de séries temporais utilizando
diferentes representações de dados e ensembles /
Rafael Giusti; orientador Gustavo Enrique de
Almeida Prado Alves Batista. -- São Carlos, 2017.
232 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, 2017.

1. Aprendizado de máquina. 2. Inteligência
artificial. 3. Séries temporais. 4. Classificação de
séries temporais. 5. Representação de séries
temporais. I. Enrique de Almeida Prado Alves
Batista, Gustavo, orient. II. Título.

Rafael Giusti

Time series classification using multiple representations and ensembles

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: Prof. Dr. Gustavo Enrique de Almeida Prado
Alves Batista

USP – São Carlos
October 2017

AGRADECIMENTOS

Este trabalho jamais teria sido possível sem o apoio de muitas pessoas. Agradeço primeiramente à minha família, por mais de três décadas de convivência e aprendizado, pelos momentos felizes e pelo suporte nos períodos mais difíceis.

Agradeço ao meu orientador, Gustavo Batista, que mais uma vez me recebeu para conduzirmos um projeto de pesquisa. Foram longos e proveitosos anos de trabalho e ensinamentos que muito estendem os sete capítulos que aqui seguem.

Agradeço à Professora Maria Carolina Monard, que me ofereceu a oportunidade de realizar meu primeiro trabalho de Iniciação Científica e me transmitiu meus primeiros valores acadêmicos. Espero poder levar adiante os seus ensinamentos ao longo da minha carreira e um dia transmiti-los a meus próprios orientandos.

Agradeço aos meus amigos e colegas do Laboratório de Inteligência Computacional. O Labic tem sido a minha segunda casa há anos e esse ambiente jamais teria sido o mesmo sem vocês. Aos que me acompanharam nestes últimos meses: Renan, Diego (Dieguito), Vinícius, Roberta, Jorge, Camila, Ivone, Antonio, Alan e Celso (êta, mininu!); aos parceiros da “velha guarda” que já partiram em novos caminhos: Bruno, Rafael (Sapão), Thiago, Ricardo, Igor, Marcos, Newton, Cristiano, Everton e Fabiano; e à nova “geração” de Labiquianos: João, Diego (Minatel), Vinícius (Ferreira), Fabiana e Ricardo (Puma).

Agradeço aos meus amigos e colegas que não estavam no Labic enquanto eu conduzia esta pesquisa, mas que fizeram parte da minha trajetória nesses últimos anos: Thyago, Lilian e Matheus (Caneca).

Agradeço à Professora Solange Rezende por todo o apoio e pelos trabalhos que realizamos no estágio PAE. Também agradeço ao Professor Robson Cordeiro, à Professora Maria das Graças Nunes, à Professora Maria da Graça Pimentel e ao Professor Marcelo Criscuolo pelos trabalhos de estágio e monitoria. Aos funcionários e docentes do Instituto de Ciências de Matemáticas e de Computação, em especial ao Serviço de Pós-Graduação que tantas vezes me atendeu quando eu tinha problemas urgentes e sem solução! À receptiva equipe do STI e ao setor de convênios, que me auxiliaram sempre que precisei.

Agradeço à Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), cujo apoio foi indispensável para a realização desta pesquisa e condução dos meus estudos.

Por fim, agradeço aos membros da Comissão Julgadora da Tese por suas correções e sugestões que serão de imensa valia em meus trabalhos futuros.

RESUMO

GIUSTI, R. **Classificação de séries temporais utilizando diferentes representações de dados e ensembles**. 2017. 232 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, 2017.

Dados temporais são ubíquos em quase todas as áreas do conhecimento humano. A área de aprendizado de máquina tem contribuído para a mineração desse tipo de dados com algoritmos para classificação, agrupamento, detecção de anomalias ou exceções e detecção de padrões recorrentes, dentre outros. Tais algoritmos dependem, muitas vezes, de uma função capaz de expressar um conceito de similaridade entre os dados. Um dos mais importantes modelos de classificação, denominado 1-NN, utiliza uma função de distância para comparar uma série temporal de interesse a um conjunto de referência, atribuindo à primeira o rótulo da série de referência mais semelhante.

Entretanto, existem situações nas quais os dados temporais são insuficientes para identificar vizinhos de acordo com o conceito associado às classes. Uma possível abordagem é transportar as séries para um domínio de representação no qual atributos mais relevantes para a classificação são mais claros. Por exemplo, uma série temporal pode ser decomposta em componentes periódicas de diferentes frequências e amplitudes. Para muitas aplicações, essas componentes são muito mais significativas na discriminação das classes do que a evolução da série ao longo do tempo.

Nesta Tese, emprega-se diversidade de representações e de distâncias para a classificação de séries temporais. Com base na escolha de uma representação de dados adequada para expor as características discriminativas do domínio, pode-se obter classificadores mais fiéis ao conceito-alvo. Para esse fim, promove-se um estudo de domínios de representação de dados temporais, visando identificar como esses domínios podem estabelecer espaços alternativos de decisão. Diferentes modelos do classificador 1-NN são avaliados isoladamente e associados em *ensembles* de classificadores a fim de se obter classificadores mais robustos.

Funções de distância e domínios alternativos de representação são também utilizados neste trabalho para produzir atributos não temporais, denominados atributos de distâncias. Esses atributos refletem conceitos de vizinhança aos exemplos do conjunto de treinamento e podem ser utilizados para treinar modelos de classificação que tipicamente não são eficazes quando treinados com as observações originais. Nesta Tese mostra-se que atributos de distância permitem obter resultados compatíveis com o estado-da-arte.

Palavras-chave: Aprendizado de máquina, Inteligência artificial, Séries temporais, Classificação de séries temporais, Representação de séries temporais.

ABSTRACT

GIUSTI, R. **Time series classification using multiple representations and ensembles**. 2017. 232 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, 2017.

Temporal data are ubiquitous in nearly all areas of human knowledge. The research field known as machine learning has contributed to temporal data mining with algorithms for classification, clustering, anomaly or exception detection, and motif detection, among others. These algorithms oftentimes are reliant on a distance function that must be capable of expressing a similarity concept among the data. One of the most important classification models, the 1-NN, employs a distance function when comparing a time series of interest against a reference set, and assigns to the former the label of the most similar reference time series.

There are, however, several domains in which the temporal data are insufficient to characterize neighbors according to the concepts associated to the classes. One possible approach to this problem is to transform the time series into a representation domain in which the meaningful attributes for the classifier are more clearly expressed. For instance, a time series may be decomposed into periodic components of different frequency and amplitude values. For several applications, those components are much more meaningful in discriminating the classes than the temporal evolution of the original observations.

In this work, we employ diversity of representation and distance functions for the classification of time series. By choosing a data representation that is more suitable to express the discriminating characteristics of the domain, we are able to achieve classification that are more faithful to the target-concept. With this goal in mind, we promote a study of time series representation domains, and we evaluate how such domains can provide alternative decision spaces. Different models of the 1-NN classifier are evaluated both isolated and associated in classification ensembles in order to construct more robust classifiers.

We also use distance functions and alternative representation domains in order to extract non-temporal attributes, known as distance features. Distance features reflect neighborhood concepts of the instances to the training samples, and they may be used to induce classification models which are typically not as efficient when trained with the original time series observations. We show that distance features allow for classification results compatible with the state-of-the-art.

Keywords: Machine learning, Artificial Intelligence, Time series, Time series classification, Time series representation.

LISTA DE ILUSTRAÇÕES

Figura 1 – Exemplo de classificação nos domínios temporal e espectral	27
Figura 2 – Esquema geral da proposta deste trabalho	29
Figura 3 – Exemplo de serialização de um contorno em forma de série temporal.	35
Figura 4 – Distância entre subsequência e série temporal.	37
Figura 5 – Demonstração do procedimento de normalização.	38
Figura 6 – Duas séries fora de fase.	42
Figura 7 – Exemplo de oclusão em séries temporais.	43
Figura 8 – Alinhamento invariante a <i>warping</i> entre duas séries temporais.	44
Figura 9 – Exemplo de invariância a escala uniforme.	44
Figura 10 – Ilustração de complexidade de séries temporais.	45
Figura 11 – Ilustração do fenômeno da distorção por complexidade.	45
Figura 12 – Estimativa da complexidade de três séries	46
Figura 13 – Séries prototípicas do conjunto CBF.	58
Figura 14 – Árvore de decisão para o conjunto tênis	73
Figura 15 – O princípio dos k vizinhos mais próximos	76
Figura 16 – Ilustração da obliquidade em torno da média.	78
Figura 17 – Exemplo de reamostragem e <i>holdout</i>	85
Figura 18 – Ilustração do processo de reamostragem na validação de classificadores.	87
Figura 19 – Exemplo de gráfico de espalhamento.	91
Figura 20 – Exemplo de gráfico <i>sharpshoot</i>	93
Figura 21 – Gráfico de distância crítica para o exemplo da Tabela 5.	98
Figura 22 – <i>Scores</i> de viés para quatro conjuntos do repositório UCR.	101
Figura 23 – Gráfico CD comparando o classificador k -NN com vizinhanças de $k = 1$ a $k = 9$	102
Figura 24 – Gráfico CD comparando o classificador k -NN com vizinhanças de $k \in \{1, 3, 4, 5\}$	102
Figura 25 – Gráficos de espalhamento comparando métodos de normalização.	103
Figura 26 – Gráficos <i>sharpshoot</i> comparando métodos de normalização.	104
Figura 27 – Raio unitário em diferentes espaços L^p	111
Figura 28 – Diferença de p na distância entre séries temporais.	111
Figura 29 – Acurácia do 1-NN utilizando distância χ^2 quadrada com normalização diferentes técnicas de normalização.	116
Figura 30 – Exemplo de <i>warping</i> de uma série sobre outra.	118

Figura 31 – Exemplo de uso de memoização no cálculo da distância DTW.	119
Figura 32 – Exemplo de <i>warping</i> patológico.	121
Figura 33 – Demonstração da banda de Sakoe-Chiba e do paralelogramo de Itakura. . .	121
Figura 34 – Exemplo de alinhamento DTW entre duas séries.	123
Figura 35 – Curvas para a função de peso logística.	125
Figura 36 – Gráfico de diferença crítica para as funções de distância no domínio temporal.	132
Figura 37 – Gráfico de distância crítica para distâncias elásticas e não elásticas.	133
Figura 38 – Desempenho do classificador 1-NN com seleção de distâncias.	134
Figura 39 – Série temporal artificial com duas componentes senoidais.	140
Figura 40 – Curvas de onda que representam as fases do sistema elétrico brasileiro. . . .	142
Figura 41 – Representação FS da série mostrada na Figura 39-esquerda.	143
Figura 42 – Exemplo de utilização da representação FS para expor sazonalidade.	143
Figura 43 – Ilustração da ausência de informação de tempo no espectro de frequências. .	144
Figura 44 – Espectrogramas obtidos com transformada “curta” de Fourier.	145
Figura 45 – Representação de uma série no domínio PAA.	150
Figura 46 – Indexação de segmentos fracionados na transformação PAA.	151
Figura 47 – Representação de uma série no domínio SAX.	154
Figura 48 – Exemplo de transformação PCA.	155
Figura 49 – Gráfico de autocorrelação para um processo estacionário e um não estacionário.	157
Figura 50 – Ilustração do modelo de máquina de vetores de suporte.	160
Figura 51 – Ilustração do <i>kernel trick</i>	160
Figura 52 – Gráfico CD para as representações com distância euclidiana.	163
Figura 53 – Gráficos de espalhamento para PCA contra ACF e SAX.	164
Figura 54 – Gráficos de espalhamento para seleção de distância contra distância euclidiana.	165
Figura 55 – Desempenho do 1-NN com distâncias elásticas nos domínios FS e ACF. . . .	166
Figura 56 – Comparação de atributos de distância com o 1-NN DTW.	168
Figura 57 – Comparação de atributos de distância e DTW com classificadores selecionados.	168
Figura 58 – Gráfico de distância crítica para os métodos de classificação avaliados neste experimento. A diferença crítica de <i>ranks</i> médios é 0,51.	169
Figura 59 – Erro do <i>ensemble</i> em função do número de classificadores base.	173
Figura 60 – Gráfico de distância crítica para todos os métodos de composição de votos. .	185
Figura 61 – Acurácia do <i>ensemble</i> dinâmico.	186
Figura 62 – <i>Ensemble</i> de acurácias e seleção de distâncias.	187
Figura 63 – <i>Ensemble</i> completo contra <i>ensemble</i> de acurácias.	188
Figura 64 – <i>Ensemble</i> completo, com seleção de distâncias, contra <i>ensemble</i> de acurácias, sem seleção de distâncias.	188
Figura 65 – Gráfico CD para o <i>ensemble</i> majoritário ponderado por acurácia.	225
Figura 66 – Gráfico CD para o <i>ensemble</i> de composição de <i>rankings</i> de probabilidade a posteriori.	225

Figura 67 – Gráfico CD para o <i>ensemble</i> majoritário ponderado por probabilidade a posteriori.	225
Figura 68 – Gráfico CD para o <i>ensemble</i> baseado em <i>simple rank</i>	225

LISTA DE ALGORITMOS

3.1	Indutor de modelo de classificação <i>Naive Bayes</i>	71
3.2	Indutor de árvores de decisão C4.5	75
3.3	Classificador k vizinhos mais próximos	77
3.4	Indução de árvores de <i>shapelets</i>	82
4.1	Distância <i>Dynamic Time Warping</i>	120
4.2	Distância LCSS	127
4.3	Distância de edição (EDR)	129
5.1	Extração de atributos de distância.	159
6.1	Variação k -NN para <i>rankings</i>	181

LISTA DE TABELAS

Tabela 1 – Descrição dos conjuntos de dados do repositório UCR.	54
Tabela 2 – Comparação de estratégias de normalização.	60
Tabela 3 – Conjunto de dados tênis.	67
Tabela 4 – Exemplo de <i>ranking</i> no teste de Wilcoxon	95
Tabela 5 – Exemplo de <i>ranks</i> em um teste de Friedman.	97
Tabela 6 – Valores críticos para o teste de Nemenyi.	98
Tabela 7 – Termos indefinidos ou não numéricos no cálculo das funções de distância.	116
Tabela 8 – Distâncias entre pares das observações das séries mostradas Figura 34.	124
Tabela 9 – Espaço de busca de parâmetros relacionados às distâncias elásticas.	130
Tabela 10 – Sumário: média e desvio padrão de acurácia de distâncias.	131
Tabela 11 – Sumário: média e desvio padrão de acurácia de distâncias (segundo experimento).	132
Tabela 12 – Pontos de corte para a representação SAX	153
Tabela 13 – Descrição dos espaços de busca dos parâmetros das representações.	161
Tabela 14 – Melhores acurácias e parâmetros para um subgrupo de conjuntos nas representações PCA e ACF.	162
Tabela 15 – Acurácias médias de distâncias e representações sobre todos os conjuntos do repositório UCR.	163
Tabela 16 – Testes de hipótese das distâncias contra a distância euclidiana	164

SUMÁRIO

1	INTRODUÇÃO	25
1.1	Contextualização	25
1.2	Hipótese e objetivos	28
1.3	Contribuições deste trabalho	29
1.4	Organização desta Tese	30
2	SÉRIES TEMPORAIS	33
2.1	Considerações iniciais	33
2.2	Conceitos	33
2.3	Comparação entre séries	36
2.3.1	<i>Conceitos associados a distância e semelhança</i>	40
2.4	Invariâncias temporais	41
2.5	Aplicações usando séries temporais	46
2.5.1	<i>Classificação de Séries Temporais</i>	47
2.5.2	<i>Previsão de Séries Temporais</i>	50
2.5.3	<i>Agrupamento</i>	52
2.5.4	<i>Detecção de motivos e anomalias</i>	53
2.6	Avaliação experimental	54
2.6.1	<i>Conjuntos de dados de benchmark</i>	54
2.6.2	<i>Procedimentos de normalização</i>	59
2.7	Considerações finais	61
3	APRENDIZADO DE MÁQUINA	63
3.1	Considerações iniciais	63
3.2	Classificação supervisionada	65
3.3	Linguagem de descrição dos dados	66
3.4	Linguagem de descrição do modelo	68
3.4.1	<i>Modelos probabilísticos de Bayes</i>	68
3.4.2	<i>Modelos de árvore de decisão</i>	72
3.4.3	<i>Modelos de vizinhança</i>	74
3.5	Indução de classificadores de dados temporais	76
3.5.1	<i>Extração de atributos</i>	77
3.5.2	<i>Classificação baseada em instâncias</i>	80

3.5.3	<i>Árvores de Shapelets</i>	81
3.6	Erro e acurácia de um classificador	83
3.6.1	<i>Validação por reamostragem</i>	87
3.6.2	<i>Validação cruzada</i>	88
3.7	Comparação de indutores	89
3.7.1	<i>Comparação por meio de gráficos</i>	91
3.7.2	<i>Teste de hipótese</i>	94
3.8	Ajuste de parâmetros	99
3.9	Avaliação experimental	99
3.9.1	<i>Comparação de métodos de validação</i>	99
3.9.2	<i>Número de vizinhos no k-NN</i>	101
3.9.3	<i>Métodos de normalização</i>	103
3.10	Considerações finais	104
4	DISTÂNCIAS ENTRE SÉRIES	107
4.1	Considerações iniciais	107
4.2	Atributos desejáveis de funções de distância	107
4.3	Classes de funções de distância	109
4.4	Métricas da família Minkowski	109
4.5	Outras medidas não elásticas	112
4.5.1	<i>Tratamento especial de medidas não elásticas</i>	115
4.6	Distâncias elásticas	117
4.6.1	<i>Dynamic Time Warping</i>	117
4.6.2	<i>Distância DTW com janela de warping</i>	120
4.6.3	<i>Distância DTW com pesos</i>	122
4.6.4	<i>Longest Common Subsequence</i>	125
4.6.5	<i>Distância de edição</i>	127
4.7	Avaliação experimental	128
4.7.1	<i>Ajuste de parâmetros</i>	129
4.7.2	<i>Utilizando diferentes funções de distância</i>	130
4.7.2.1	<i>“Atualização” dos experimentos</i>	132
4.7.2.2	<i>O “1-NN dinâmico”</i>	133
4.8	Considerações finais	134
5	TRANSFORMAÇÃO DE SÉRIES	137
5.1	Considerações iniciais	137
5.2	Representações baseadas em frequências	138
5.2.1	<i>Representação DFT</i>	140
5.2.2	<i>Representação espectral</i>	141
5.2.3	<i>Short-time Fourier transform</i>	144

5.2.4	<i>Transformada discreta de wavelets</i>	146
5.3	Representação PAA (<i>Piecewise Aggregate Approximation</i>)	149
5.4	Representação SAX (<i>Symbolic Aggregate approXimation</i>)	152
5.5	Componentes Principais	155
5.6	Representação autocorrelativa	156
5.7	Atributos de distância	158
5.8	Avaliação experimental	161
5.8.1	<i>Ajuste de parâmetros</i>	161
5.8.2	<i>Representações individuais</i>	162
5.8.2.1	<i>Diversidade de representações e distâncias elásticas</i>	165
5.8.3	<i>Atributos de distância</i>	166
5.9	Considerações finais	169
6	ENSEMBLES DE CLASSIFICADORES	171
6.1	Considerações iniciais	171
6.2	<i>Ensembles</i> de classificadores	172
6.3	Estratégias de geração de diversidade	174
6.3.1	<i>Bagging</i>	174
6.3.2	<i>Ensemble de representações</i>	175
6.4	Estratégias de composição de votos	176
6.4.1	<i>Voto majoritário</i>	177
6.4.2	<i>Composição de rankings</i>	180
6.5	Avaliação experimental	184
6.5.1	<i>Comparação de mecanismos de voto</i>	184
6.5.2	<i>“Atualização” dos experimentos</i>	186
6.6	Considerações finais	188
7	CONCLUSÕES	191
7.1	Resumo do projeto e revisão do trabalho	191
7.2	Principais contribuições e limitações	193
7.3	Possibilidades de trabalhos futuros	196
	REFERÊNCIAS	199
	APÊNDICE A APÊNDICE: RESULTADOS ESTENDIDOS	215
A.1	Viés e variância dos métodos de reamostragem	215
A.2	Resultados completos do teste de normalização e <i>ranks</i> de Friedman	219
A.3	Parâmetros de distâncias elásticas	221
A.4	Classificador 1-NN dinâmico	223
A.5	Gráficos de distância críticas para os <i>ensembles</i>	224

A.6	Teste do <i>ensemble</i> dinâmico	225
A.7	“Atualização” do <i>ensemble</i> de acurácias	226
A.8	Outros resultados	228

INTRODUÇÃO

1.1 Contextualização

Na inteligência artificial, a linha de pesquisa conhecida como aprendizado de máquina (AM) tem feito contribuições significativas para a análise dados temporais. O aprendizado de máquina tem como objetivo a pesquisa de sistemas capazes de aprender automaticamente. Grande esforço da comunidade científica tem sido dedicado a desenvolver algoritmos capazes de lidar com dados temporais ou adaptar técnicas de AM para esse tipo de dados. Métodos baseados em similaridade—ou, por equivalência, em distância—entre séries temporais são amplamente utilizados em praticamente todas as tarefas de mineração de dados temporais. Na detecção de exceções ou anomalias, uma função de similaridade pode ser empregada para encontrar ocorrências anômalas de algum fenômeno: uma série temporal muito distinta das demais pode ser considerada uma anomalia (YANKOV; KEOGH; REBBAPRAGADA, 2007), havendo evidências de que métodos baseados em similaridade promovem resultados em geral superiores (CHANDOLA; CHEBOLI; KUMAR, 2009). Na detecção de *motifs*, uma função de similaridade desempenha papel central para encontrar subsequências recorrentes (MUEEN *et al.*, 2009). Na previsão de séries temporais, uma função de distância pode ser utilizada para encontrar subsequências semelhantes àquela que antecede uma observação desconhecida, cujo valor pode ser estimado com base na evolução dessas subsequências (ODAN *et al.*, 2009). Na maioria das técnicas de agrupamento, incluindo hierárquicas, particionais e fractais, uma função de similaridade é crucial para detectar objetos próximos (ROKACH, 2005). E, na classificação, uma função de similaridade pode ser utilizada para classificar uma série com base nas classes das séries mais semelhantes (SILVERMAN; DIXON, 1976; AGRAWAL; FALOUTSOS; SWAMI, 1993).

O trabalho relatado nesta Tese tem como foco a *classificação de séries temporais*. Na classificação, os dados são associados a conceitos denominados *classes*. Formalmente, a classificação de dados temporais envolve o aprendizado de um mapeamento $f : \mathbb{T} \mapsto \mathbb{Y}$, no qual

$\mathbb{T}$ representa um espaço de dados sequenciais ou séries temporais e $\mathbb{Y}$ representa um espaço de classes. Todas as séries temporais $x \in \mathbb{T}$ de uma mesma classe $C_j \in \mathbb{Y}$ são conceitualmente relacionadas de alguma forma. O objetivo da classificação é determinar a classe de uma série temporal para a qual essa informação é desconhecida. O aprendizado de máquina contribui para essa tarefa com algoritmos e técnicas que possibilitam a um sistema computacional encontrar, de maneira automática, uma relação entre os dados que pode ser utilizada para discriminar as classes. Diversos problemas reais podem ser solucionados por meio de classificação de séries temporais com aplicações em várias áreas, como medicina (MELGANI; BAZI, 2008), astronomia (RICHARDS *et al.*, 2011), entomologia (BATISTA *et al.*, 2011) e biometria (SILVA *et al.*, 2012; STARNER *et al.*, 1994), dentre outras.

Resultados empíricos sugerem que busca baseada em similaridade é capaz de produzir resultados extremamente competitivos para uma vasta gama de domínios de aplicação (WANG *et al.*, 2013; DING *et al.*, 2008). Com base em uma função de similaridade e um conjunto de dados temporais representativo, uma série temporal pode ser classificada encontrando-se um subconjunto de objeto semelhantes e atribuindo-se à série desconhecida a moda das classes presentes nesse conjunto. A esse procedimento dá-se o nome de regra dos k vizinhos mais próximos (COVER; HART, 1967). O caso particular em que um único vizinho é utilizado para classificar uma série é conhecido como o classificador “1 vizinho mais próximo” ou 1-NN.

No domínio temporal, os atributos discriminativos das classes podem estar profundamente “embutidos” em relações complexas entre as observações. Por exemplo, em processamento de sinais digitais de áudio, é comum que as séries temporais apresentem um grande número de observações, sendo que cada observação configura um atributo pouco representativo. Isso faz com que funções de similaridade apresentem dificuldade em discriminar as classes e, por conseguinte, que os resultados obtidos por um classificador sejam pobres. A perda de poder de discriminação de funções de similaridade em espaços de grande dimensionalidade e na presença de muitas dimensões irrelevantes é um problema conhecido como a maldição da dimensionalidade (HOULE *et al.*, 2010).

É possível transportar as séries do domínio temporal para um domínio no qual atributos mais relevantes são evidentes. Por exemplo, uma série temporal pode ser reescrita por meio de uma decomposição em componentes periódicas no domínio espectral (AGRAWAL; FALOUTSOS; SWAMI, 1993). Nesse domínio, cada observação representa a amplitude de uma componente de determinada frequência.

A fim de ilustrar esse cenário, na Figura 1, três séries temporais de sinais de áudio são mostradas com suas representações nos domínios temporal (*esquerda*) e espectral (*direita*). As três séries foram coletadas durante o desenvolvimento de um trabalho conduzido no ICMC-USP (SILVA, 2014), no âmbito de um projeto FAPESP¹. Os detalhes do projeto e de

¹ Projeto “Sensores Inteligentes para Controle de Pragas Agrícolas e Insetos Vetores de Doenças”, realizado com apoio da Função de Amparo à Pesquisa do Estado de São Paulo (proc. 2012/50714-7).

coleta das séries podem ser consultados em (BATISTA *et al.*, 2011). Por ora, basta saber que um inseto, ao cruzar um sensor, produz dados que são gravados como sinais de áudio e que esses sinais variam de acordo com a espécie do inseto. O objetivo é identificar a espécie do inseto que produziu aquele sinal. No exemplo da figura, as classes correspondentes às três séries são conhecidas, porém o classificador não tem acesso à informação de classe da série C.

Esse exemplo ilustra de forma intuitiva como o problema da maldição da dimensionalidade afeta a classificação. Na Figura 1-esquerda, é difícil expressar matematicamente qual das duas séries, A ou B, é mais semelhante à série C. O classificador 1-NN, utilizando a distância euclidiana no domínio temporal, classificaria incorretamente a série C como a espécie *Culex quinquefasciatus*. Já na Figura 1-direita, tem-se as mesmas séries temporais reescritas no domínio espectral. A maior semelhança entre as séries temporais B e C é facilmente perceptível nesse caso, apesar da significativa diferença de amplitudes. Nesse novo cenário, o mesmo classificador concluiria corretamente que a série C corresponde a um inseto da espécie *Aedes aegypti*.

O exemplo sugere que a classificação das séries teria se tornado mais “fácil” no domínio espectral. Para problemas nos quais o discernimento das classes tem relação direta com as frequências presentes no sinal, os coeficientes da série no domínio espectral são atributos muito mais significativos para a classificação.

O domínio espectral não é o único domínio no qual séries temporais podem ser reescritas. Muitas técnicas de processamento de sinais digitais de alguma forma empregam a representação

Figura 1 – Três série temporais, obtidas a partir de três sinais de áudio, representadas nos domínios temporal – esquerda – e espectral – direita. O eixo das abcissas indica os índices das observações; o eixo das coordenadas indica os valores das observações em cada representação. A classe da série C é omitida. Algumas observações espúrias da representação espectral foram removidas

de séries temporais como espectros, cepstros ou espectrogramas (BATISTA *et al.*, 2011; HAO; CAMPANA; KEOGH, 2012; BRESOLIN; NETO; ALSINA, 2008). Para diversas outras aplicações, a comunidade de AM tem explorado domínios de representação que incluem representações simbólicas (LIN *et al.*, 2007), baseadas em *wavelets* (CHAN; FU, 1999), segmentativas (KEOGH *et al.*, 2001; CHEN *et al.*, 2007) etc.

Cada domínio promove a reescrita dos dados temporais em atributos que tornam explícita alguma “visão” das séries. Supõe-se que o uso adequado desses domínios pode tornar mais “fácil” a condução de alguma atividade de mineração de dados temporais. Nesta Tese, os termos “representação” e “diversidade de representações” serão empregados frequentemente. O objetivo principal deste trabalho é explorar as representações na busca de atributos que forneçam com maior poder discriminativo das classes em problemas de classificação de séries temporais.

1.2 Hipótese e objetivos

Esta pesquisa está sustentada na hipótese de que:

“Não existe uma única representação de séries temporais capaz de evidenciar as características discriminativas das classes para todo e qualquer domínio de aplicação.”

Assumindo que essa hipótese é verdadeira, três importantes itens de pesquisa podem ser identificados:

1. a investigação e a descrição dos principais domínios de representação;
2. o estudo desses domínios de representação e sua adequação para a classificação de séries temporais;
3. a possibilidade de se compor os resultados obtidos em diferentes domínios em um único classificador por meio de técnicas de *ensemble*.

Embora o foco principal desta Tese seja depositado na tarefa de classificação, os resultados ora discutidos potencialmente se aplicam a qualquer atividade de mineração de dados conduzida através de uma estratégia baseada em similaridade. Se uma representação é adequada para tornar explícitas as características de algum domínio de aplicação, isso pode se refletir em ganho de desempenho não apenas em classificação, mas também em agrupamento, predição de séries, detecção de anomalias etc. Note que essa colocação é meramente *assumida* como uma possível contribuição secundária deste trabalho, não sendo parte das hipóteses verificadas na pesquisa.

É importante reiterar que este trabalho não está focado em nenhuma aplicação específica, como a classificação de insetos, de dígitos falados ou de reconhecimento de escrita. O objetivo

desejado neste trabalho é contribuir para o estado-da-arte e explorar e relatar o uso de técnicas usualmente desconsideradas, idealmente encontrando domínios de representação que sejam eficazes na classificação de séries temporais ou levantando evidências de que tais domínios são em geral inferiores àqueles tipicamente empregados na classificação.

1.3 Contribuições deste trabalho

O principal objetivo desta Tese é apresentar estratégias para utilizar representações de dados temporais e funções de distância como mecanismo que torna evidentes as características discriminativas das classes. A Figura 2 mostra o princípio geral da proposta.

Figura 2 – Esquema geral da proposta deste trabalho

As principais contribuições dessa proposta, previstas no início da execução deste trabalho, são:

1. um levantamento das principais formas de representação de séries temporais e uma visão unificada dessas representações, apresentado ao longo do Capítulo 5;
2. um levantamento de funções de distância para séries temporais que possam ser utilizadas sobre as representações levantadas. Ao longo do Capítulo 4, são apresentadas diversas funções de distância. Foram selecionadas distâncias relacionadas à norma L_p , derivadas de coeficientes de similaridade para distribuições de probabilidade, baseadas em entropia de Shannon, bem como distâncias elásticas que promovem invariância a *warping* local e oclusão. Essa diversidade de distâncias é associada a diversidade de representações no que, pelo conhecimento do autor, consiste um trabalho original ausente na área de classificação de séries temporais;
3. uma proposta para a construção de *ensembles* de classificadores utilizando diferentes estratégias de construção e seleção de representações. Por exemplo, espera-se que uma associação de classificadores no domínio temporal a classificadores no domínio de frequências seja capaz de classificar novos exemplos considerando as características discriminativas de ambos os domínios de representação simultaneamente, utilizando alguma estratégia de composição de votos;

Além disso, durante a condução da pesquisa foram identificados pontos de interesse diretamente relacionados ao projeto de Tese:

1. um estudo sobre diferentes métodos de normalização de dados temporais. Normalmente, pesquisadores de classificação de séries temporais assumem que a normalização em escores padrões é a forma mais adequada de se normalizar esse tipo de dados para promover a comparação por meio de normalidade. Nesta Tese, considera-se o emprego de outras duas estratégias de normalização, apresentadas na Seção 2.3. Na Seção 3.9.3, serão apresentadas evidências que corroboram essa escolha;
2. uma avaliação de funções de distância alternativas que podem ser melhor utilizadas para classificação de séries temporais utilizando métodos de normalização alternativos. Essas distâncias são normalmente ignoradas pela comunidade de AM, possivelmente porque tipicamente produzem escores inválidos quando empregadas com séries temporais normalizadas em escores padrões. Nesta Tese, é mostrado que algumas dessas funções de distância podem produzir classificadores consideravelmente acurados quando associados a outras técnicas de normalização (Seção 4.5.1);
3. um método para extração de atributos não temporais que incorporam o conceito de vizinhança promovido por diversidade de representações e de distância. Denominados atributos de distância (Seção 5.7), esses atributos permitem utilizar o poder discriminativo de mudança de representação e diversidade de distâncias para a construção de modelos de classificação utilizando linguagens de descrição arbitrárias, como árvores de decisão, modelos probabilísticos ou vetores de suporte. A avaliação experimental sugere que os atributos de distância permitem a construção de classificadores bastante competitivos com o estado-da-arte;

Por fim, foi criado um *website* para armazenar todos os resultados de experimentos obtidos durante a realização deste projeto, bem como o *software* produzido para a condução desses experimentos (GIUSTI, 2017). Essa pode ser considerada uma contribuição técnica desta Tese. O autor acredita fortemente que todo estudo científico deve ser acompanhado de um esforço para que a reprodução dos resultados seja o mais simples possível para qualquer atuante da área.

Essas contribuições serão discutidas ao longo de toda a Tese. Na Seção 7.2, será feita uma “releitura” desses pontos. Os resultados alcançados e as recomendações para a prática de classificação de séries temporais que puderam verificadas serão revistas e discutidas. As limitações deste trabalho também serão consideradas nessa seção.

1.4 Organização desta Tese

No Capítulo 2 são abordados os conceitos de séries temporais necessários para o entendimento do restante da Tese, incluindo a notação que será utilizada no texto. A leitura desse capítulo não exige conhecimento de métodos de aprendizado de máquina; os conceitos de distância e vizinhança são introduzidos para a apresentação de conceitos relacionados. Cada um

dos capítulos seguintes, exceto a Conclusão, aborda um aspecto majoritário da execução deste projeto.

No restante desta Tese, cada capítulo está sub-estruturado da seguinte maneira.

- *Considerações iniciais*: a primeira subseção de cada capítulo tem como objetivo contextualizar o tema do capítulo no projeto e possivelmente apresentar hipóteses levantadas nesta Tese;
- *Discussão*: apresentam os conceitos necessários para compreensão do trabalho realizado e revisão bibliográfica (múltiplas seções);
- *Avaliação experimental*: a penúltima subseção de cada capítulo é dedicada a apresentar experimentos realizados para verificar as hipóteses levantadas no início do capítulo, comparar estratégias relevantes para o tema e para explicar e justificar eventuais conclusões empíricas apresentadas nas seções anteriores;
- *Considerações finais*: na última seção de cada capítulo, são discutidos os resultados obtidos e apresentadas conclusões sobre o tema do capítulo.

Especificamente, no Capítulo 3 são introduzidos conceitos de aprendizado de máquina de modo geral e para séries temporais. É feita uma discussão sobre os desafios em aprendizado de máquina pertinentes à classificação de dados temporais. O algoritmo k -NN é apresentado na Seção 3.4.3 como um classificador de propósito geral. Na Seção 3.6.1 são abordadas estratégias de validação de modelos, com especial destaque para a validação cruzada. Na Seção 2.6.1 são apresentados os conjuntos de dados que serão utilizados em todas as avaliações experimentais seguintes.

A partir do Capítulo 4, o classificador k -NN é dissociado da distância euclidiana. As principais funções de distância empregadas neste trabalho são apresentadas nas primeiras seções do capítulo. Os experimentos deste capítulo têm por objetivo mostrar como diferentes funções de distância afetam o classificador k -NN para diferentes domínios de aplicação.

A partir do Capítulo 5, o classificador k -NN é novamente generalizado e dissociado do domínio temporal. Domínios de representação alternativos de séries temporais são abordados e resultados experimentais são apresentados para mostrar como diferentes domínios de representação podem ser empregados na classificação por similaridade. Na Seção 5.7, mostra-se como funções de distância e representações alternativas podem ser combinadas para extrair atributos que podem ser utilizados com diferentes modelos de classificação. Os experimentos apresentados nesse capítulo têm por objetivo mostrar principalmente como a noção de vizinhança pode ser incorporada em modelos de classificação com linguagens de descrição arbitrárias. Os experimentos mostrados nesse capítulo utilizam como modelo de teste as máquinas de vetores de suporte.

No Capítulo 6, as representações e funções de distância estudadas nos capítulos anteriores são combinadas em *ensembles* de classificadores. São avaliadas diferentes estratégias de construção de *ensembles* e de composição de votos.

Finalmente, no Capítulo 7, são apresentados resumos dos principais conceitos utilizados neste trabalho e das principais conclusões atingidas nos capítulos anteriores. As limitações deste trabalho, incluindo os tópicos de pesquisa não explorados e questões em aberto são discutidos e possíveis trabalhos futuros são abordados.

SÉRIES TEMPORAIS

2.1 Considerações iniciais

O TEMPO É UM ELEMENTO UBÍQUO na experiência humana. Praticamente todas as atividades que podemos realizar e os fenômenos que podemos observar trazem consigo alguma componente de natureza temporal. Em muitos casos, o estudo dessa componente é essencial para que seja possível aprimorar a forma como uma atividade é realizada ou para que se possa melhor compreender algum aspecto da natureza.

Uma forma de representar dados temporais é através de séries temporais monovariadas, que são, sucintamente, coleções de observações sequenciais que expressam o desencadeamento de algum fenômeno durante um período de tempo. Séries temporais são um tipo de dado que é explorado em diversas áreas do conhecimento, como entomologia, administração de recursos públicos, astronomia etc. Uma aplicação particularmente relevante para esta Tese é a classificação de séries temporais, na qual o objetivo é identificar um conceito associado a um objeto temporal.

Neste capítulo, são explorados os conceitos fundamentais relacionados a séries temporais necessários para a execução deste trabalho. Nas Seções 2.2 e 2.4, são apresentadas as definições de série temporal, subsequência e distância que serão utilizadas ao longo da Tese. Uma revisão da literatura de séries temporais e aplicações que utilizam esse tipo de dados é apresentada na Seção 2.5. Na Seção 2.6 são apresentados resultados experimentais referentes às hipóteses levantadas ao longo do capítulo e, finalmente, na Seção 2.7 são apresentados um resumo e considerações finais dos conceitos abordados.

2.2 Conceitos

Séries temporais são um caso particular de dados temporais (FU, 2011). Podemos definir uma **série temporal** como um conjunto de observações quantitativas ordenadas cronologica-

mente (KIRCHGÄSSNER; WOLTERS, 2007a; CHATFIELD, 2004). Séries temporais podem implicitamente representar conceitos muito mais complexos do que a simples coleção de suas observações e, por isso, são naturalmente empregadas para descrever fenômenos que variam com o tempo. Muitas vezes o estudo de séries temporais permite compreender melhor a natureza dos fenômenos por elas descritos. O físico alemão A. Schuster é habitualmente apontado como o autor de um dos trabalhos acadêmicos mais antigos a empregar séries temporais no estudo de fenômenos da natureza (SCHUSTER, 1906). Tendo como base dados coletados ao longo de 150 anos de observação do Sol, Schuster empregou técnicas de análise de séries temporais para propor o modelo cíclico de 11 anos que atualmente é entendido como a melhor explicação da atividade solar.

Formalmente, uma série temporal pode ser definida como a *realização* de um processo estocástico. Um **processo estocástico** é um processo aleatório que se desenvolve cronologicamente. Mais precisamente, um processo estocástico é um conjunto ordenado de variáveis aleatórias $\mathbf{X} = \{\mathbf{X}_t\}$. O índice t é contido em $\mathbb{N} = \{1, 2, 3, \dots\}$ quando o processo ocorre a intervalos discretos no tempo e contido em $t = [0, \infty[$ quando ocorre continuamente (LAWLER, 2006).

Definição 2.1: *Um processo estocástico de tempo discreto é um conjunto contável de variáveis aleatórias $\mathbf{X} = \{\mathbf{X}_1, \mathbf{X}_2, \dots\}$, tais que $\mathbf{X}_t : \Omega \mapsto \mathbb{R}$ descrevem os possíveis estados do processo ao longo do tempo.*

No escopo desta Tese, séries temporais são obtidas a partir da amostragem de algum fenômeno ao longo do tempo. Por exemplo, um indivíduo pode lançar uma moeda repetidamente e anotar os resultados dos lances. A sequência de observações obtida é uma realização de um processo estocástico de tempo discreto. Embora esse tipo de processo seja potencialmente infinito, o indivíduo está limitado a algum número arbitrário de observações, tanto em tempo quanto espaço. Esse tipo de restrição se aplica a qualquer fenômeno que pode ser observado e amostrado digitalmente, portanto podemos desconsiderar do escopo desta Tese séries temporais de tempo contínuo ou infinitas:

Definição 2.2: *Uma série temporal é uma sequência finita de observações $S = (s[1], s[2], \dots, s[n])$, tomadas a partir da amostragem de um processo estocástico de tempo discreto.*

Fenômenos contínuos são representados como processos estocásticos de tempo discreto através de um procedimento denominado **amostragem**. O número de amostras tomadas durante o processo de amostragem é denominado **taxa de amostragem** da série temporal. Por exemplo, um sinal digital de áudio é uma representação discreta do som e tipicamente contém 44.100 amostras por segundo. Portanto, a taxa de amostragem típica de sinais de áudio é de 44.100 Hz ou 44,1 kHz.

Definição 2.3: *A taxa de amostragem f_s de uma série temporal é o número de observações tomadas a cada unidade de tempo.*

A duração de uma série temporal é o produto da taxa de amostragem pelo número de observações amostradas, $f_s \cdot n$ ¹.

Definição 2.4: Seja $S = (s[1], s[2], \dots, s[n])$ uma série temporal amostrada a uma taxa f_s . Diz-se que o tamanho, a dimensionalidade ou a cardinalidade de S é o número de observações $|S| = n$ e sua duração é $n \cdot f_s$, com unidade compatível com f_s .

Séries temporais podem ser utilizadas na representação de dados não temporais por meio de um “truque de representação” (KEOGH *et al.*, 2009), o qual denominaremos **serialização**. O procedimento para serializar um objeto ou uma forma pode ser explicado mais adequadamente por meio de um exemplo. Considere o crânio mostrado na Figura 3-a, um objeto tridimensional complexo, do qual um perfil é obtido e projetado sobre uma superfície, formando uma curva fechada (b). Para obtermos a serialização desse perfil, escolhemos um ponto de início e percorremos um circuito em sentido anti-horário (c), amostrando valores ordenadamente. Por exemplo, a série mostrada na parte inferior da figura foi obtida por um processo de serialização no qual o ponto inicial foi definido como a parte mais projetada da mandíbula. As observações da série foram amostradas como a distância de cada ponto do circuito ao centro de gravidade do perfil.

Figura 3 – Exemplo de serialização de um contorno em forma de série temporal—figura adaptada de (KEOGH *et al.*, 2009).

Existe um aspecto temporal relacionado ao processo de serialização, implícito na ideia de que o circuito em torno de uma forma geométrica, figura bidimensional ou projeção é feito sequencialmente durante um intervalo de tempo. Entretanto, esse aspecto temporal é irrelevante para o objeto amostrado em si. Em vez disso, o conceito fundamental que se deseja representar pela série temporal é a ordem entre as observações.

No restante desta Tese, o termo “série” será empregado como sinônimo de “série temporal”. O termo **subsequência** será utilizado para indicar um subconjunto ordenado de observações contíguas de uma série².

¹ O operador $\cdot$ será empregado nesta Tese para indicar multiplicação quando a notação $x \cdot y$ for preferível à xy .

² Em outros domínios, uma subsequência de elementos contíguos é tipicamente denominada *substring*.

As notação adotada nesta Tese é a seguinte:

- Uma série será identificada como uma letra romana maiúscula—tipicamente S ou Z , mas não se limitando a estas—e suas observações serão minúsculas romanas equivalentes, indexadas com colchetes—*e.g.*, $F = (f[1], f[2], \dots, f[n])$;
- A notação $s[a:b]$ será utilizada para indicar uma subsequência de S entre $s[a]$ e $s[b]$, incluindo esses extremos. Isto é, $s[a:b] = (s[a], s[a+1], \dots, s[b])$. Está implícito em $s[a:b]$ que a relação $1 \leq a \leq b \leq |S|$ é verdadeira. Para todos os efeitos, uma subsequência pode ser empregada em qualquer contexto no qual uma série seria válida;
- Séries integrantes de conjuntos serão representadas com a mesma notação empregada para séries e observações, podendo estas serem acrescidas de um índice subscrito para discriminar elementos do conjunto. Isto é, o conjunto $D = \{S_1, S_2, \dots, S_N\}$ contém N séries temporais, $S_1 = (s_1[1], s_1[2], \dots, s_1[n_1])$, $S_2 = (s_2[1], s_2[2], \dots, s_2[n_2])$ e assim por diante;
- A mesma extensão poderá ser utilizada para descrever uma subsequência de uma série temporal pertencente a um conjunto. Isto é, se S_i é uma série temporal de D , então $s_i[a:b]$ é uma subsequência de S_i .

Note que um conjunto de séries temporais não precisa ser limitado a séries de mesma dimensão. Assim, o conjunto D pode residir em um espaço de vetores não homogêneos. O espaço de todas as possíveis séries temporais será representado como $\mathbb{T}$. O conjunto de todas as séries de tamanho n é simplesmente $\mathbb{R}^n$.

2.3 Comparação entre séries

Diversas aplicações de séries temporais envolvem identificar o grau de semelhança entre duas ou mais séries. Em muitos casos, isso pode ser obtido pela distância euclidiana. A distância euclidiana para séries temporais retém o mesmo conceito que a distância euclidiana em um espaço de vetores e é dada pela Equação 2.1.

$$d_{\text{Euc}}(S, Z) = \sqrt{\sum_{t=1}^n |s[t] - z[t]|^2} \quad (2.1)$$

Em algumas aplicações, é necessário encontrar a distância entre uma subsequência e uma série temporal. Não existe uma definição única desse tipo de distância. Neste trabalho, adotou-se o conceito mais utilizado em classificação de séries temporais por similaridade, que é a menor

O conceito de subsequência adotado nesta Tese está de acordo com o que é utilizado na literatura de classificação de séries temporais. Uma exceção à essa definição será observada na Seção 4.6.4, na qual trataremos da distância LCSS.

distância entre a subsequência e todas as subsequências que podem ser encontradas na série completa.

Definição 2.5: *Seja $S = (s[1], s[2], \dots, s[n])$ uma série temporal. Considere a subsequência $z[1:m]$ extraída de uma série temporal Z qualquer (possivelmente $Z = S$) tal que $m \leq n$. A distância euclidiana entre a subsequência e a série temporal pode ser definida como a menor distância entre a subsequência de interesse e todas as subsequências de S .*

$$d_{SubSeq}(z[1:m], S) = \min_{t=0}^{n-m} d_{Euc}(z[1:m], s[t+1:t+m]). \quad (2.2)$$

Por exemplo, um *shapelet* é uma subsequência representativa de um conceito (YE; KEOGH, 2011). Como será explicado na Seção 3.5.3, a distância entre uma série e um *shapelet* pode ser utilizada para construir árvores de decisão. De acordo com a Definição 2.5, uma proximidade entre a série e a subsequência indica que os eventos descritos pelo *shapelet* (ou eventos semelhantes) podem ser observados em algum instante ao longo da série. Na Figura 4, mostra-se a comparação de uma subsequência com uma série: a primeira subsequência destacada ao longo da série é a mais próxima da subsequência de interesse; a outra, por contraste, é a mais distante. A distância entre a subsequência e a série é definida pela mais próxima, isto é, $d_{SubSeq} = 0,693$.

Figura 4 – Distância entre subsequência e série temporal. A subsequência de interesse é comparada com todas as subsequências da série. A distância entre a subsequência e a série é dada pela subsequência mais próxima ($d = 0,693$). Em contraste, a subsequência mais distante também é destacada na figura ($d = 7,637$).

Uma desvantagem da distância euclidiana é sua sensibilidade a variações no eixo do tempo e no eixo da amplitude. A distância euclidiana não consegue capturar adequadamente a similaridade de séries próximas que variam apenas em magnitude (Figura 5-a) ou *offset* (Figura 5-b). Porém isso pode ser facilmente contornado através da **normalização** das séries. Na classificação de séries temporais, o objetivo da normalização é representar todas as séries de um conjunto em uma escala comum sem alterar os critérios discriminativos das classes. Isso pode ser obtido, por exemplo, com a **normalização em escores padrões**, também conhecida como **z-normalização** (KEOGH; KASETTY, 2003; HU; CHEN; KEOGH, 2013). Um exemplo desse processo está exemplificado na Figura 5-c.

Figura 5 – Demonstração da importância de normalização. Nos três painéis, são exibidos os mesmos pares de séries, porém construídos de forma a simular divergências de amostragem. Em *a*, as séries foram escaladas e divergem em amplitude. Em *b*, uma das séries foi “deslocada” para cima a fim de simular divergência de *offset*. Em *c* apresenta-se as séries normalizadas em escores padrões. A distância euclidiana para cada par de séries é, respectivamente: 7,29; 8,73 e 5,24.

A normalização em escores padrões ou *z*-normalização consiste em substituir os valores de uma amostra pelos seus escores padrões. Estritamente, o escore padrão deve ser obtido com base nos parâmetros de média e desvio padrão da população—parâmetros que, em aplicações reais, são normalmente desconhecidos. Entretanto, consideremos que o objetivo da *z*-normalização na classificação por similaridade é simplesmente transladar e redimensionar as séries para que todas tenham média zero e desvio padrão unitário. Portanto os escores podem ser calculados com base nos valores amostrados da média e do desvio padrão de cada série.

Definição 2.6: *Seja $S = (s[1], s[2], \dots, s[n])$ uma série temporal. A função de *z*-normalização é um mapeamento $\mathfrak{z} : \mathbb{R}^n \mapsto \mathbb{R}^n$ que substitui todas as observações de S pelos escores*

$$\mathfrak{z}(S) = (z[1], z[2], \dots, z[n]), \quad (2.3)$$

na qual cada $z[t]$ corresponde ao escore da t -ésima observação

$$z[t] = \frac{s[t] - m}{s}, \quad (2.4)$$

considerando-se a média m e o desvio padrão s de S :

$$m = \sum_{t=1}^n \frac{s[t]}{n} \quad \text{e} \quad (2.5)$$

$$s = \sqrt{\sum_{t=1}^n \frac{(s[t] - m)^2}{n - 1}}. \quad (2.6)$$

Apesar de simples, a função de *z*-normalização exige um cuidado, pois uma série pode ser constante ($s[1] = s[2] = \dots = s[n]$). Como a variância dessa série é $s^2 = 0$, a função $\mathfrak{z}$ se torna indefinida. Outra situação pode ocorrer quando uma série é aproximadamente constante, porém possui um pequeno número de observações que pouco diferem da média—*e.g.*, $S =$

$(1; 1; 1; 1; 1,0001; 1)$. A pequena diferença entre a quinta e as demais observações de S pode ser super-expressa pela z -normalização. Isso é especialmente importante quando são analisadas subsequências de séries, pois a probabilidade de se encontrar subsequências constantes e quase-constantes é relativamente alta em algumas aplicações (LIN *et al.*, 2007). Além disso, devido às características práticas da aritmética de ponto flutuante, nem sempre a variância de uma série ou subsequência constante será exatamente zero. Nesses casos, pode-se testar a variância estimada contra um ponto de corte ε . Se $s^2 < \varepsilon$, então utiliza-se $s = 1$ na Equação 2.4. No trabalho desta Tese, adota-se $\varepsilon = 10^{-10}$.

A normalização em escores padrão não é, entretanto, a única estratégia de normalização que pode ser adequada para classificação de séries temporais por similaridade. Por exemplo, uma série pode ser normalizada para o intervalo $[0, 1]$ substituindo-se a relação $z[t]$, na Equação 2.4, por

$$u[t] = \frac{s[t] - \min}{\max - \min}, \quad (2.7)$$

no qual $\min$ e $\max$ referem-se, respectivamente, às observações de menor e maior valor da série S . Uma série pode também ser normalizada como um histograma se for normalizada para o intervalo $[0, 1]$ e então escalonada para que a soma de suas observações seja unitária. Isto é, substituindo-se a relação $z[t]$ na Equação 2.4 por

$$h[t] = \frac{u[t]}{\sum_{i=1}^n s[i]}. \quad (2.8)$$

Note que uma série temporal “normalizada como um histograma” não é propriamente um histograma. Essa expressão é utilizada nesta Tese apenas para indicar que as observações da série apresentam algumas propriedades compatíveis com as de um histograma—*i.e.*, uma curva limitada em $[0, 1]$ cuja área é unitária.

Em todo caso, sempre que alguma função de normalização é utilizada sobre as séries de um conjunto de dados, é comum que a mesma função seja empregada na definição de distância entre uma subsequência e uma série. Por exemplo, quando se utiliza z -normalização, substitui-se a relação (2.2) na Definição 2.5 por

$$d_{\text{SubSeq}}(z[1:m], S) = \min_{t=0}^{n-m} d_{\text{Euc}}(z[1:m], \mathfrak{z}(s[t+1:t+m])). \quad (2.9)$$

Outras medidas de distância podem ser utilizadas no lugar da função euclidiana (Capítulo 4). Algumas delas exigem que a série seja normalizada como um histograma. Porém, a z -normalização é tipicamente considerada um padrão *de fato* para normalização de séries temporais. Embora não seja uma questão “fechada”, muitos autores consideram que normalizar em escores padrões é “essencial” para qualquer aplicação em que o conceito de similaridade entre séries temporais é essencial para a construção de um modelo ou hipótese (LIN *et al.*, 2007).

Finalmente, é importante considera-se também as propriedades do **lower bounding**. Em muitas aplicações, pode ser interessante obtermos rapidamente uma estimativa da distância mínima entre duas séries.

Definição 2.7: *Sejam S e Z séries temporais. Uma função de lower bounding, para uma função de distância $d(\cdot, \cdot)$, é um mapeamento $f : \mathbb{T} \times \mathbb{T} \mapsto \mathbb{R}$ tal que*

$$f(S, Z) \leq d(S, Z). \quad (2.10)$$

Esse escore mínimo da distância entre duas séries pode ser utilizado, por exemplo, para encontrar subsequências recorrentes em uma série temporal de alta dimensionalidade. Em vez de calcularmos a distância entre todos os pares de subsequências, podemos utilizar a função de *lower bounding* para rapidamente eliminar todos os pares cuja distância seja superior a um limiar ε . Isso permite reduzir o espaço de busca e a complexidade assintótica no caso médio.

2.3.1 Conceitos associados a distância e semelhança

Ao longo desta Tese, serão utilizados termos como “proximidade”, “semelhança”, “casamento” e “vizinhança”. Tais termos expressam conceitos propositalmente vagos e são utilizados na literatura em situações nas quais uma definição formal seria desnecessariamente difícil ou tediosa. Por exemplo, pode-se dizer que duas séries são *próximas* ou *semelhantes* quando o valor de alguma função de distância entre elas é “pequeno”. Ou quando se deseja remeter à intuição através de uma inspeção visual de forma ou correlação das séries.

Um conceito bem-definido para subsequências é o de **sobreposição**. Duas subsequências, $s[i:i+n]$ e $s[j:j+m]$, ambas “extraídas” de uma mesma série temporal S e convenientemente indexadas de modo que $i \leq j$, sobrepõem-se se $i + n \geq j$. Já o termo *casamento* pode possuir diferentes sentidos dependendo da forma como é empregado. Em alguns contextos, trata-se de encontrar pares de séries cuja distância entre elas seja inferior a um limiar r . Em outros contextos, trata-se de encontrar séries ou subsequências “próximas”—por exemplo, a distância entre uma subsequência e uma série é definida nesta Tese como o melhor casamento mais adequado entre elas. Já um *casamento trivial* entre duas subsequências ocorre quando a sobreposição entre elas é superior a algum número de observações.

Outro conceito bem definido é o de vizinho mais próximo. Dado um conjunto de séries temporais D , o **vizinho mais próximo** de uma série S é a série $Z \in D$ tal que não existe nenhum $Z' \in D$ com $d(S, Z') < d(S, Z)$. Já o termo *vizinhança* é muitas vezes empregado de forma vaga. A vizinhança de uma série S pode referir-se tanto a um subconjunto de D contendo séries semelhantes a S quanto ao hiperespaço no qual residem todas as possíveis séries próximas de S .

Ao longo desta Tese, sempre que for adequado e claro, estes e outros termos serão utilizados de maneira intuitiva, sem definição formal prévia.

2.4 Invariâncias temporais

Diz-se que uma função de distância é **invariante** a alguma característica indesejada dos dados se ela permite comparação *robusta* entre séries temporais mesmo na presença dessa característica. Por exemplo, em aplicações envolvendo classificação de áudio, dizemos que uma distância é invariante a ruído se ela for capaz de ignorar pequenas distorções causadas por imprecisão nos sensores, som ambiente ou qualquer outro tipo de dado aleatório que não acrescenta informação útil ao sinal. De acordo com (BATISTA; WANG; KEOGH, 2011), cada domínio de aplicação pode exigir suporte a um conjunto específico de invariâncias, de modo que a escolha de uma função de distância adequada seria a etapa mais importante na classificação de séries temporais. Por exemplo, dados provenientes de captura de movimento tipicamente são afetados por variações locais e não lineares de aceleração, o que acarreta em uma distorção denominada *warping* local.

Batista, Wang e Keogh (2011) listam sete tipos de características potencialmente indesejáveis, para as quais são propostas invariâncias:

Amplitude e offset: desvios em amplitude e *offset* são provavelmente as distorções mais comuns do processo de amostragem. A presença desse tipo de distorção em bases de dados temporais pode fazer com que séries com formas semelhantes sejam consideradas anormalmente distintas, como mostrado na Figura 5 (pág. 38). Porém é possível obter-se invariância a esses fenômenos trivialmente por meio da normalização da série. Nesta Tese, são considerados três procedimentos para normalizar séries temporais. Alguns autores defendem que obter invariância a amplitude e *offset* é essencial para a classificação por similaridade. Por exemplo, Lin *et al.* (2007) argumentam que a classificação por similaridade de séries temporais não normalizadas “não faz sentido”.

Fase: o conceito de *fase* diz respeito à “posição” em que uma onda é observada durante seu ciclo (BALLOU, 2013). Mais especificamente, a *diferença de fase* diz respeito ao estado da onda no momento em que ela é inicialmente observada. Séries temporais podem diferir em fase principalmente quando o fenômeno amostrado não possui um instante inicial claro. Por exemplo, existem sistemas destinados a constantemente observar e anotar a luminosidade de estrelas, produzindo séries temporais. Essas séries tipicamente se encontram *fora de fase* (KEOGH *et al.*, 2009; PROTOPAPAS *et al.*, 2006), como mostrado na Figura 6-*superior*. Como a distância euclidiana não é invariante a fase, essa distorção causa a impressão de que as séries são mais distintas do que na realidade. Ao rotacionar uma das séries e ajustar suas fases, a distância euclidiana reflete a distância verdadeira entre as séries (Figura 6-*inferior*).

Podemos obter invariância a fase de forma trivial com um algoritmo de força bruta. Isto é, calculamos a distância de uma série de tamanho n para todas as suas possíveis $n - 1$ versões

Figura 6 – Na parte *superior* da figura, as duas séries estão fora de fase. Na parte *inferior*, a série de traço fino (vermelha) foi rotacionada para ter sua fase sincronizada com a outra série (BATISTA; WANG; KEOGH, 2011).

rotacionadas e escolhemos a rotação que produz a menor distância. Esta é, entretanto, uma abordagem computacionalmente cara e inviável em grande parte das aplicações.

Quando soluções aproximadas são aceitáveis, uma opção é extrair atributos invariantes a rotação (CARDONE; GUPTA; KARNIK, 2003)—como coeficientes de Fourier, área sob a curva, entropia etc.—e realizar a classificação apenas com esses atributos. Porém, uma solução não trivial e mais eficiente do que o método de força bruta pode ser obtida com a seguinte observação: o algoritmo de força bruta é um processo iterativo no qual rotações cada vez melhores (mais próximas da original) são obtidas. Quando o algoritmo encontra a rotação ideal, ele passa a verificar rotações “inferiores”. A proposta de Keogh *et al.* (2006) é utilizar uma função de *lower bounding* para rapidamente obter limites inferiores da distância euclidiana. Se o limite inferior de uma rotação é maior do que a menor distância encontrada até a iteração atual, então essa rotação é “inferior” e a distância real não precisa ser calculada. Além disso, os autores propõem ordenar o espaço de rotações de modo a verificar primeiramente as rotações mais “promissoras”. Apesar de ter a mesma complexidade de tempo assintótica que o método de força bruta, o ganho médio em tempo de execução, relatado pelos autores, é de duas ordens de magnitude.

Oclusão: essa invariância é necessária quando subsequências da série podem estar ausentes ou conter informação que não representa o fenômeno observado. Isso pode acontecer quando o processo de amostragem é propenso a erros ou quando o objeto de amostragem é incompleto. Na Figura 7, as séries são amostradas a partir do contorno de crânios humanos. É natural que as amostras arqueológicas tenham sofrido danos e sejam incompletas. Como se observa na figura, essas falhas se manifestam no processo de serialização. A subsequência destacada na figura deve ser tratada como um caso especial, pois não representa com fidelidade o seio nasal de um crânio intacto.

Pode-se obter invariância a oclusão por meio de um *envelope* em torno de uma série—representado na Figura 7 pela área cinza. Nesse caso o limite superior da série S é obtido

Figura 7 – Exemplo de oclusão em séries temporais: alguns pedaços do crânio na parte inferior da figura foram perdidos por desgaste natural. A série temporal amostrada pelo contorno desse crânio possui subsequências faltantes e o processamento desses dados exige invariância a oclusão (BATISTA; WANG; KEOGH, 2011).

aplicando-se um *offset* $+\epsilon$ e o limite inferior por um *offset* $-\epsilon$. As observações de Z que estejam “fora” do envelope de S são ignoradas. Na Figura 7, o alinhamento mostrado é produzido pela distância LCSS (*Longest Common Subsequence*).

Escala local: o *warping* local é um fenômeno bastante comum que se apresenta como distorções localizadas no eixo temporal. Aqui, o termo “localizado” refere-se ao fato de que diferentes segmentos da série podem sofrer maior ou menor efeito de *warping*. Diversos fatores podem contribuir para esse tipo de distorção, como erros de amostragem ou a simples ausência de uniformidade temporal do processo amostrado. Por exemplo, séries temporais podem ser utilizadas para representar caligrafia (RATH; MANMATHA, 2003). É razoável e intuitivo supor que pequenas variações no traço surjam naturalmente do processo de escrita, especialmente quando se estuda a caligrafia de diferentes pessoas.

A distância *Dynamic Time Warping* (DTW) pode ser utilizada para promover comparação invariante a *warping* local. Ela altera a escala do eixo temporal de maneira não uniforme e promove um alinhamento não linear entre as séries, como exemplificado na Figura 8. A distância DTW é abordada na Seção 4.6.1 desta Tese.

Escala global: essa invariância é obtida quando se reescala uma série de forma uniforme no eixo temporal. Invariância a escala global ou uniforme é necessária, por exemplo, em aplicações de *query by humming*³, em que o usuário deseja identificar uma música “murmurando” apenas um trecho de sua melodia (EUACHONGPRASIT; RATANAMAHATANA, 2008). Hu, Chen e Keogh (2013) defendem que qualquer aplicação real de séries temporais deve

³ Neste contexto, *humming* é o ato de “cantar de lábios fechados”.

Figura 8 – Alinhamento invariante a *warping* entre duas séries temporais.

partir do pressuposto de que as séries são coletadas de maneira não uniforme e que esse tipo de invariância é sempre necessário.

Um exemplo de aplicação de escala global é apresentado na Figura 9. Deseja-se verificar se a série de interesse Q é uma subsequência provável da série candidata C . À esquerda, mostra-se o melhor casamento entre a série candidata e a série de interesse. Quando a série de interesse Q é submetida a uma escala uniforme de fator 1,2, o melhor casamento é indicado na Figura 9-direita. Como se pode observar pela figura, a distância entre as séries é bastante reduzida quando comparada com invariância a escala global.

Figura 9 – Exemplo de invariância a escala uniforme. A reta tracejada indica o novo tamanho da série de consulta (Q) quando submetida a uma escala uniforme de fator 1,2.

No exemplo da Figura 9, o fator de escala necessário para se obter invariância a escala era conhecido. Entretanto, no caso típico, é necessário verificar diferentes fatores até que se encontre o mais apropriado. Esse processo pode ser eficientemente realizado quando uma medida de *lower bounding* é utilizada para reduzir o espaço de busca (KEOGH *et al.*, 2003).

Complexidade: invariância a complexidade é um conceito relativamente recente (PRATI; BATISTA, 2013; BATISTA *et al.*, 2013). A importância desse tipo de invariância pode ser demonstrada por meio de um exemplo artificial. Na Figura 10, os contornos de seis formas geométricas com diferentes números de vértices são convertidos em séries temporais.

Quanto maior o número de vértices, mais complexa a série resultante, como pode ser observado pelo número de picos e vales das séries na parte inferior da figura.

Figura 10 – Seis contornos de diferentes complexidades foram amostrados em séries temporais. A complexidade de cada série é consistente com a complexidade das figuras associadas—fonte: (BATISTA; WANG; KEOGH, 2011).

Esse tipo de invariância normalmente não é necessário quando os dados possuem graus próximos de complexidade. Porém, quando um conjunto de dados possui séries que diferem muito em complexidade, uma série complexa tende a ser considerada mais próxima de séries simples—e vice-versa. A intuição por trás desse fenômeno é que a série complexa tende a “oscilar” em torno da série simples, minimizando a distância média entre as observações, enquanto séries complexas possuem picos e vales que tendem a desalinhar. Um exemplo real é mostrado na Figura 11, na qual a distância entre séries que representam folhas de espécies distintas é menor que a distância entre séries de mesma espécie.

Figura 11 – Ilustração do fenômeno da distorção por complexidade: a série mais simples, referente a um espécime de *B. cylindrica*, é considerada mais próxima da série complexa, referente a um espécime de *A. artemisiifolia* (parte superior) do que outro espécime de mesma espécie cujo contorno produz uma série complexa (parte inferior)—fonte: (BATISTA; WANG; KEOGH, 2011).

Batista *et al.* (2013) introduziram o conceito de **fator de correção de complexidade**: um valor de escore atribuído a pares de séries dependendo da complexidade relativa entre elas.

Pares de séries igualmente complexas têm fator de correção $CF=1$. O fator de correção é maior para séries com maior diferença de complexidade. Assim, a distância euclidiana invariante a complexidade é definida como na Equação 2.11.

$$d_{CID}(S, Z) = d_{Euc} \cdot CF_{SZ} \quad (2.11)$$

Na prática, o efeito introduzido por essa correção é tornar a distância mais “exigente” com pares de séries de diferentes complexidades. A utilidade dessa correção depende de um método adequado de estimar a complexidade das séries (PRATI; BATISTA, 2012; PARMEZAN, 2016). A estratégia proposta por Batista *et al.* (2013) consiste em considerar as séries como cadeias de segmentos de reta entre as observações. Quanto maior a complexidade de uma série, maior o comprimento de cada um desses segmentos. Portanto, se esses segmentos forem alinhados, o resultado é uma linha reta que tende a ser mais longa para séries mais complexas (fixado o número de observações). Esse conceito é mostrado na Figura 12.

Figura 12 – Estimativa da complexidade de três séries. A série menos complexa, quando “esticada”, produz a sequência mais curta. A série mais complexa produz a sequência mais longa—(BATISTA; WANG; KEOGH, 2011)

É de se esperar que, em uma aplicação real, as séries temporais estejam sujeitas a diferentes tipos de distorções, o que pode exigir o uso de múltiplas técnicas de invariâncias. O caso mais comum é o emprego da distância DTW em séries temporais normalizadas—situação na qual se toma proveito de invariância à amplitude, *offset* e *warping* local. *Ensembles* de classificadores (Capítulo 6) também podem ser utilizados para combinar diferentes modelos de classificação e, potencialmente, diferentes invariâncias.

2.5 Aplicações usando séries temporais

Séries temporais são encontradas nos mais variados domínios de aplicação, englobando diferentes tipos de problemas e exigindo diferentes estratégias de solução. A área de aprendizado de máquina contribui para esses domínios de maneiras diversas. As principais tarefas envolvendo séries temporais nas quais é usual empregar aprendizado de máquina são as seguintes.

- **Classificação:** cada série temporal é produzida por um fenômeno distinto ou representa uma classe distinta de objetos. Dada uma série temporal, o objetivo é descobrir qual é o fenômeno gerador ou qual classe de objetos ela representa;

- *Agrupamento*: dado um conjunto de séries temporais, o objetivo é encontrar uma estrutura natural que permita distribuir as séries em grupos;
- *Previsão*: uma série temporal, potencialmente infinita, representa algum fenômeno de interesse. O objetivo da previsão é estimar o valor de uma ou mais observações futuras de um fenômeno parcialmente realizado;
- *Detecção de motivos*: também denominado detecção de *motifs*; o objetivo é encontrar uma ou mais subsequências recorrentes ao longo da série;
- *Detecção de anomalias*: encontrar subsequências ou séries que são inesperadas em algum contexto.

Nas próximas subseções, são apresentados trabalhos relacionados nos quais os problemas supracitados são encontrados em aplicações reais.

2.5.1 Classificação de Séries Temporais

Na **classificação** de séries temporais, assume-se que as séries são associadas a um rótulo semântico denominado **classe**. A classe pode ser o fenômeno gerador da série ou pode estar associada ao objeto que produziu a série—*e.g.*, serialização. O objetivo do aprendizado de máquina em tarefas de classificação é desenvolver um modelo de conhecimento que seja capaz de identificar a classe associada a uma série temporal com base nas suas observações. Este é o principal problema envolvendo séries temporais abordado nesta Tese e será apresentado em mais detalhes no Capítulo 3.

O escopo de problemas que podem ser abordados por meio de classificação de séries temporais é vasto. Algumas aplicações tradicionais envolvem a classificação de sinais digitais, como áudio. Em (NOLL, 1967), atributos derivados da representação de frequências foram utilizados para identificar tons em gravações de voz. O avanço da área levou do reconhecimento de tons para o reconhecimento de conteúdo. Bresolin, Neto e Alsina (2008) utilizaram transformação de *wavelets* para extrair coeficientes de séries obtidas de gravações de dígitos pronunciados em português—“zero”, “um”, “dois” etc. Utilizando máquinas de vetores de suporte, os autores classificaram as séries temporais de acordo com o dígito pronunciado. Ainda no contexto de classificação de voz, Hamooni e Mueen (2014) focaram em identificar os fonemas—a menor unidade inteligível de som, a partir das quais sílabas e então palavras são construídas. Os autores empregaram um classificador misto que utiliza distância DTW para classificar uma série temporal de acordo com uma taxonomia hierárquica de fonemas e, em seguida, coeficientes baseados em *wavelets* para identificar o fonema específico. Em uma abordagem um pouco mais indireta, Hao, Campana e Keogh (2012) produziram espectrogramas (representação visual das componentes de frequências de uma série ao longo do tempo) de sons produzidos por insetos e empregaram uma medida de distância entre imagens para classificar as espécies dos insetos.

Uma área das ciências “puras” que é bastante beneficiada por métodos de análise de séries temporais é a astronomia. Atualmente, existem muitos sistemas autônomos que compilam terabytes de dados temporais a partir de sensores ópticos ou de rádio. Alguns são destinados a mensurar o brilho de corpos celestes. A mensuração desse brilho ao longo do tempo produz séries temporais denominadas *curvas de luzes*. Diversas aplicações na astronomia envolvem classificar curvas de luzes. [Debosscher et al. \(2007\)](#) extraíram atributos periódicos dessas séries e utilizaram um classificador baseado em modelos de mistura gaussiana ([REYNOLDS, 2015](#)) para identificar as denominadas *estrelas variáveis*, que são estrelas cuja intensidade do brilho varia substancialmente em períodos relativamente curtos. Dentre as diversas explicações para o brilho variável de uma estrela, existe o caso especial no qual ela é orbitada por um planeta, caracterizada por ciclos nos quais a intensidade da luz diminui brevemente quando ocorre um trânsito⁴. Até o século XX, apenas algumas dezenas de exoplanetas eram conhecidos. Em 2017, o número total de exoplanetas confirmados já era superior a 3 mil ⁵. Uma dificuldade ainda presente é o alto número de classes de estrelas variáveis ([BLOMME et al., 2010](#)) e a dificuldade em classificá-las devido ao ruído presente nos dados temporais ([RICHARDS et al., 2011](#)). É de grande interesse da comunidade científica, portanto, que técnicas de aprendizado de máquina que potencializem a classificação dessas séries sejam exploradas e refinadas.

Muitos exames médicos produzem volumes extremamente elevados de dados temporais ([WANG et al., 2013](#)). Um exemplo clássico é o eletroencefalogramas (EEG), obtido da leitura contínua de sensores posicionados em diferentes regiões do crânio. O EEG é uma importante ferramenta para revelar padrões de funcionamento do cérebro e é utilizado no diagnóstico e tratamento de diversas patologias, como a epilepsia ([FISHER et al., 2014](#); [SMITH, 2005](#)). A epilepsia é um conjunto de disfunções neurológicas caracterizadas por convulsões recorrentes. Episódios convulsivos são classificados em quatro estágios: pré-convulsivo, pós-convulsivo, intra-convulsivo e, no caso de diagnóstico de epilepsia, inter-convulsivo⁶. Dados temporais extraídos de EEG ([PETROSIAN, 1995](#)) ou de inspeção visual ([TZALLAS; TSIPOURAS; FOTIADIS, 2007](#)) podem ser utilizados para classificar a situação físico-mental do paciente nessas quatro categorias de modo automático. Essa classificação é de particular relevância porque, embora a medicina tradicionalmente aceite que convulsões são “imprevisíveis”, evidências recentes sugerem correlações entre esses estágios, o que permitiria prever convulsões em pacientes que já receberam um diagnóstico de epilepsia. [Cook et al. \(2013\)](#) estudaram os efeitos médicos envolvidos na instalação de sensores intra-craniais que poderiam ser utilizados para realizar essa classificação, com o intuito futuro de produzir dispositivos para advertir pacientes de um episódio convulsivo iminente. Em 2014 e 2106 a universidade australiana de Melbourne liderou

⁴ Um trânsito é caracterizado quando um corpo celeste parece “passar a frente” de outro, do ponto de vista terrestre.

⁵ *The Extrasolar Planets Encyclopaedia*: <<http://exoplanet.eu>>.

⁶ Os termos da área médica são pré-ictal, pós-ictal, ictal e inter-ictal, respectivamente e dizem respeito a padrões encontrados no EEG, podendo referir-se não apenas a convulsões epiléticas, mas também convulsões não epiléticas, derrames e enxaquecas.

a realização de competições que encorajam a comunidade científica a pesquisar e desenvolver classificadores para esse fim⁷.

Outro exame bastante comum na área médica que produz dados temporais é o eletrocardiograma (ECG). O exame de ECG compartilha com o EEG o fato de utilizar sensores elétricos para detectar de forma não intrusiva o funcionamento de um órgão interno. Nesse caso, o objetivo é identificar os padrões rítmicos dos batimentos cardíacos. Possivelmente a contribuição mais relevante de AM nesse escopo é a identificação dos principais pontos de referência desse ritmo (WILLEMS *et al.*, 1987), que pode ser dividido em estágios: contração dos átrios, relaxamento dos átrios, contração dos ventrículos e relaxamento dos ventrículos⁸. Cardiologistas segmentam o ECG em ondas que correspondem aos impulsos elétricos associados a esses estágios. O segmento mais característico é provavelmente o *complexo QRS*, que corresponde aos impulsos elétricos relacionados ao relaxamento dos ventrículos. Diversos trabalhos de AM procuram segmentar e classificar essas subsequências. Andreão, Dorizzi e Boudy (2006) utilizaram modelos de Markov para esse fim e produziram sequências anotadas dos segmentos, a partir das quais procuram identificar e classificar os batimentos como normais ou ventricular-prematurados⁹. Outro trabalho relevante é o de (SAINI; SINGH; KHOSLA, 2013), no qual filtros de frequências foram empregados para remover componentes potencialmente irrelevantes para o complexo QRS. As séries filtradas foram processadas e convertidas em curvas de gradientes, das quais foram obtidas subsequências. Dentre outras abordagens, os autores utilizaram o classificador *k*-NN (Seção 3.4.3) para identificar subsequências relacionadas ao complexo QRS.

Séries temporais podem ser utilizadas também no combate a vetores transmissores de doenças—como o *Aedes aegypti*, potencial portador de vírus que causam quatro tipos distintos de febre: dengue, amarela, zica e chicungunha. Uma das etapas fundamentais no combate a qualquer vetor é estimar a densidade da sua população. Tendo esse fim em mente, Batista *et al.* (2011) propuseram utilizar sensores ópticos para capturar padrões de batidas de asas como séries temporais e extrair atributos de frequências para treinar classificadores capazes de identificar a espécie e o sexo do inseto. Souza (2016) empregou estratégias de agrupamento que permitem adaptar, de modo não supervisionado, um modelo de classificação a mudanças nos conceitos das classes em fluxos de dados temporais. A relevância desse trabalho ao combate de vetores se deve ao fato de que o comportamento dos insetos é bastante influenciado por variações ambientais, como mudanças de temperatura. Lemes (2016) empregou algoritmos de classificação *anytime* para classificar séries temporais. A relevância de algoritmos *anytime* advém nessa aplicação advém da necessidade de se utilizar dispositivos de baixo custo e poder computacional que possam ser posicionados em locais remotos e em grande número. Como

⁷ Edição de 2016: <www.kaggle.com/c/melbourne-university-seizure-prediction>.

⁸ O coração humano normal possui quatro cavidades: as duas menores são os átrios; as duas maiores são os ventrículos.

⁹ Batimentos caracterizados pela contração prematura de um ventrículo, tipicamente associados a arritmia.

consequência, a frequência com que os insetos cruzam o leitor pode ser elevada demais para que o sistema seja capaz de processar totalmente todos os exemplos. O classificador *anytime* admite interrupções do processo de classificação buscando apresentar soluções intermediárias que não são substancialmente inferiores (em termos de precisão ou acurácia, por exemplo) àquelas que seriam encontradas caso o classificador não fosse interrompido.

Classificação de séries temporais pode ser empregada também em situações nas quais a relação temporal não é explícita. Por exemplo, na avaliação da qualidade de alimentos, emprega-se técnicas de espectroscopia. Em síntese, espectroscopia consiste em utilizar ondas eletromagnéticas para estudar a composição e as propriedades de algum objeto (NAWROCKA; LAMORSKA, 2013), podendo ser utilizado, por exemplo, para detectar adulteração de carnes (AL-JOWDER; KEMSLEY; WILSON, 2002). Ondas eletromagnéticas são casos claros e bastante conhecidos de séries temporais e o espectrograma, derivado dessas ondas, pode ser visto como um caso especial de uma série temporal representada no domínio de amplitudes (Seção 5.2.2). Espectrogramas podem também ser utilizados para avaliar a qualidade de óleo de cozinha e vinho (BAGNALL *et al.*, 2012).

Um exemplo mais drástico de classificação de séries temporais em domínios não temporais se dá quando as séries são amostradas a partir dos contornos de objetos (WEI; KEOGH; XI, 2006). Esse procedimento, explicado e exemplificado na Seção 2.2 (pág. 35), foi utilizado em (HEARN, 2009) para representar folhas (vegetais) como sequências ordenadas de distâncias da borda ao centro. Com base na distância euclidiana das representações espectrais dessas sequências, o autor realizou classificação por similaridade para identificar as espécies das folhas. Um conceito semelhante é o de *perfis de palavras*. Dada a forma escrita de uma palavra, seu perfil é uma sequência de pontos extraídos do seu contorno de forma tal que compõem uma série. Rath e Manmatha (2003) extraíram perfis de palavras encontradas em textos caligrafados e utilizaram distância DTW para classificar as diferentes palavras—*e.g.*, “Alexandria”, “letters” etc. Outros trabalhos mostram como séries temporais podem ser utilizadas para classificar diversas famílias de formatos, como crânios (KEOGH *et al.*, 2006), artefatos pré-históricos (XI *et al.*, 2007) e insetos (HILLS *et al.*, 2014).

2.5.2 Previsão de Séries Temporais

Em séries temporais, **previsão** consiste em estimar os valores de observações ainda não coletadas. Isto é, dada uma série temporal referente a um fenômeno parcialmente observado $S = (s[1], \dots, s[n])$, deseja-se estimar o valor da observação $s[n+1]$ antes que esta seja amostrada—ou, em uma variação mais difícil, a subsequência $s[n+1:n+t]$.

Algumas das aplicações mais conhecidas de previsão de séries temporais são encontradas na economia, na sociologia e na biologia. Modelos de recorrência como *SARIMA* podem ser ajustados aos dados para capturar componentes de tendência, sazonalidade e resíduo (PARMEZAN, 2016). A previsão é realizada extrapolando-se o intervalo de coleta e simulando observações

futuras.

Esse tipo de projeção é feito, por exemplo, para estimar o consumo futuro de algum recurso limitado. Kam, Sung e Park (2010) utilizaram modelos SARIMA para representar o número diário de pacientes que visitam um pronto-socorro. Com base em modelos desse tipo, é possível estimar a quantidade de pacientes que irá recorrer ao serviço com alguns dias de antecedência, permitindo à unidade de atendimento preparar-se para a demanda (KIM *et al.*, 2014).

Malhas viárias modernas são equipadas com sensores que constantemente coletam dados de tráfego urbano ou inter-urbano. Alguns autores possuem interesse em utilizar esses dados para *estimar* o tempo médio das viagens que já foram realizadas. Em outros casos, o questionamento consiste em *prever* qual será a duração de uma viagem no momento em que ela se inicia. De acordo com (CARRASCAL, 2015), ambas as perguntas envolvem aproximar dados faltantes. O autor sugere uma situação na qual o problema de estimativa pode ser reduzido a um problema de previsão de séries temporais quando sensores fornecem apenas dados para um segmento inicial da viagem.

Um trabalho que envolve princípios semelhantes aos adotados nesta Tese foi conduzido por (FERRERO, 2009). Utilizando dados de sensores instalados na barragem de Itaipu, o autor empregou um algoritmo baseado no princípio dos k vizinhos mais próximos para prever riscos de falhas estruturais na barragem. Destaca-se nesse trabalho a introdução do algoritmo k -NN-TSP no lugar de modelos regressivos. Para prever uma observação, o autor seleciona uma subsequência imediatamente anterior a esta e procura as subsequências mais próximas no histórico da série. As observações que sucedem esses “vizinhos” são utilizadas para estimar o valor da observação desconhecida.

Uma aplicação associando redes sociais e economia foi empregada por Bollen, Mao e Zeng (2011) na plataforma de *microblogging* Twitter¹⁰. Muitos usuários do Twitter compartilham mensagens que refletem seu humor—*e.g.*, “estou feliz”. Os autores compilaram uma base de mensagens desse tipo e as classificaram como expressões de diferentes sentimentos—*e.g.*, “felicidade” e “tranquilidade”. Com base nessas mensagens, os autores produziram séries temporais que refletem a intensidade média desses sentimentos entre os usuários da plataforma ao longo de aproximadamente 9,5 meses. Primeiramente, os autores verificaram que essa séries fornecem informação sócio-cultural observando, por exemplo, as subsequências das séries relacionadas a feriados públicos e na proximidade das eleições nacionais de 2008. Em seguida, os autores utilizaram essas séries como informação contextual para diferentes modelos de previsão treinados para prever séries do índice de ações DJIA¹¹. A conclusão apresentada em (BOLLEN; MAO; ZENG, 2011) é que a informação temporal extraída da plataforma Twitter pode ser utilizada para

¹⁰ Rede social de compartilhamento de mensagens limitadas a 140 caracteres—<www.twitter.com>.

¹¹ O Dow Jones *Industrial Average* (DJIA) é um índice médio de empresas do setor industrial—<<http://www.djaverages.com/?go=industrial-overview>>.

melhorar a precisão na previsão do valor de fechamento das ações.

2.5.3 Agrupamento

Agrupar é o processo de encontrar **grupos** ou *clusters* naturais em um conjunto de dados (ESLING; AGON, 2012). Os elementos de cada grupo devem ser semelhantes entre si—*i.e.*, deve haver baixa variância intra-classe—e distintos dos exemplos pertencentes aos outros grupos—*i.e.*, alta variância inter-classe. É semelhante à classificação no sentido de que busca separar os elementos em categorias, com a importante distinção de que classes são conceitos inerentes aos exemplos, enquanto grupos são associações post-hoc impostas sobre os dados. O objetivo da classificação é decidir a qual classe um exemplo pertence (REZENDE, 2003), enquanto o objetivo do agrupamento é encontrar estruturas (potencialmente desconhecidas) associadas à distribuição dos dados (LIAO, 2005).

O problema de agrupamento é tipicamente abordado de forma incremental. Inicialmente, assume-se a existência de um ou mais grupos rudimentares, possivelmente aleatórios. A cada iteração, esses grupos são refinados. Um dos algoritmos mais conhecidos é o *k*-médias (HAN; PEI; KAMBER, 2011), que iterativamente encontra centroides para cada grupo dos dados (JAIN, 2010). Outros algoritmos produzem agrupamento *hierárquico*, o que se assemelha a estabelecer uma espécie de taxonomia aos dados (XIAO; LUO; GAO, 2012).

De modo geral, técnicas de agrupamento são tipicamente utilizadas como ferramentas para auxiliar na inspeção dos dados (AGHABOZORGI; SHIRKHORSHIDI; WAH, 2015). Em particular, é interessante descobrir grupos em domínios nos quais as semânticas associadas não são inteiramente conhecidas a ponto de se determinar classes inerentes aos exemplos. Para ilustrar essa aplicação, considere o trabalho de (GENOLINI *et al.*, 2016), no qual os autores se propuseram a classificar séries temporais de acordo com os formatos *das séries*—não devendo ser confundido com séries que *representam formas*. Dois estudos de caso foram apresentados nesse trabalho. No primeiro, os autores argumentaram que a estratégia proposta revelou a existência de um grupo de pacientes de Alzheimer que até então era desconhecido. No segundo, os autores alegaram terem encontrado comportamentos hormonais do organismo feminino durante o ciclo hormonal menstrual que contrariam a literatura. Outro exemplo é encontrado em (WIJK; SELOW, 1999), no qual os autores propuseram uma estratégia de visualização de grupos em séries temporais e, como estudo de caso, mostraram como o agrupamento pode ser utilizado para inferir padrões desconhecidos de comportamento dos trabalhadores de uma empresa.

Agrupamento pode ser utilizado também como técnica para auxiliar a realização ou demonstração de outras tarefas de AM. Muitos autores apresentam hierarquias de grupos para mostrar de forma intuitiva, por exemplo, como o conceito de vizinhança varia com a escolha de uma função de distância ou representação de séries temporais (KUMAR *et al.*, 2005; LIN; KHADE; LI, 2012). Como ferramenta auxiliar, destaca-se a utilidade de agrupamento na classificação de fluxos de dados. Fluxos de dados (*data streams*) são notáveis devido à alta incidência

de mudanças de conceito. Em síntese, mudança de conceito ocorre quando as características discriminativas das classes se tornam distintas daquelas levantadas na construção do modelo. Nessas situações, é importante que o modelo de classificação seja alterado a fim de acompanhar essa mudança nas características discriminativas—caso contrário, o modelo pode se tornar progressivamente mais impreciso. O trabalho de (SOUZA, 2016) é de particular importância porque assume diferentes cenários nos quais o conceito de um fluxo de dados muda ao longo do tempo e propõe utilizar agrupamento como forma de “acompanhar” mudanças na distribuição dos dados. De posse dessa informação, é possível adaptar o modelo de classificação e minimizar a perda de qualidade dos resultados.

2.5.4 Detecção de motivos e anomalias

Anomalias e motivos são conceitos próximos tipicamente associados a subsequências de séries temporais. Motivos (ou *motifs*) são padrões frequentes encontrados em uma sequência ou série temporal (MALETZKE, 2009; YANKOV *et al.*, 2007), enquanto anomalias são padrões infrequentes. O que exatamente determina se uma subsequência ou série é frequente depende da formulação do problema. Alguns autores definem motivos como a subsequência cujo casamento não trivial possui menor distância dentre todas as subsequências da série. Já o conceito de k -ésimo motivo estabelece que o primeiro motivo é a subsequência que possui o maior número de casamentos não triviais com distância máxima r —portanto o k -ésimo motivo é o primeiro motivo que não possui casamento trivial com os $k - 1$ motivos mais significativos. Para anomalias ou exceções, a denominação geral estabelece que uma série temporal é anômala quando é produzida por um processo distinto do que seria esperado para uma classe ou conjunto de séries temporais (CHANDOLA; CHEBOLI; KUMAR, 2009). Por exemplo, em um estudo de caso envolvendo eletrocardiogramas, Kumar *et al.* (2005) identificaram como anomalias um conjunto de séries que foram posteriormente confirmadas por um especialista como sendo na verdade registros de células marca-passo normais¹².

Outro caso específico de anomalias em séries temporais são os denominados *discords* (KEOGH; LIN; FU, 2005). Um termo em português para *discord* poderia ser “dissonante”. Dada uma série temporal S de tamanho $|S| = n$, existem $n - m + 1$ subsequências em S de tamanho $m \leq n$. Para uma subsequência $x = s[i:i+m-1]$ qualquer de S , há $n - 2m + 1$ outras subsequências de S cujo casamento com x é não trivial. Estas são denominadas subsequências vizinhas de x . Uma subsequência dissonante de tamanho m é aquela cuja distância para a subsequência vizinha mais próxima é máxima dentre todas as subsequências de tamanho m de S .

Na astronomia, uma tarefa relevante é identificar *outliers* em curvas de luzes. Protopapas *et al.* (2006) consideraram o problema de detectar *outliers* dentre curvas de luzes obtidas de

¹² Refere-se, neste caso, às células presentes no coração que ditam o ritmo cardíaco, não ao dispositivo homônimo que reproduz artificialmente o comportamento dessa célula.

estrelas variáveis periódicas. Os autores calcularam a correlação média entre pares de séries no domínio de frequências e as séries menos correlacionadas com seus pares foram selecionadas como potenciais *outliers*. Outra área na qual *outliers* entre séries é empregada é a mecânica, na qual séries temporais representam características estruturais de componentes maquinários e *outliers* podem indicar a existência de peças defeituosas (YANKOV; KEOGH; REBBAPRAGADA, 2008).

2.6 Avaliação experimental

Os experimentos relacionados a este capítulo foram conduzidos com o objetivo de validar duas hipóteses levantadas nas seções anteriores: a necessidade de se utilizar um ponto de corte no processo de z -normalização e a importância e a diferença entre métodos de normalização de séries temporais.

2.6.1 Conjuntos de dados de benchmark

Os experimentos desta Tese foram conduzidos principalmente nos conjuntos de dados do repositório UCR (CHEN *et al.*, 2015). Oficialmente denominado “*UCR Time Series Classification Archive*”, esse repositório hospedado pela Universidade da Califórnia, campus de Riverside (UCR), possui atualmente 85 conjuntos de dados temporais com características bastante distintas e englobam vários domínios de aplicação. Trata-se, potencialmente, do maior repositório aberto de dados temporais rotulados para classificação e tem sido extensivamente utilizado em publicações científicas nos anos recentes.

Na Tabela 1 são listados os conjuntos de dados do repositório UCR e suas principais características. Todos os conjuntos são z -normalizados e particionados em exemplos de treinamento e teste.

Tabela 1 – Descrição dos conjuntos de dados do repositório UCR.

Conjunto de dados	Tipo	Classes	Exemplos de Treino	Exemplos de Teste	Observações
50Words	Serialização	50	450	455	270
Adiac	Serialização	37	390	391	176
ArrowHead	Serialização	3	36	175	251
Beef	Espectrograma	5	30	30	470
BeetleFly	Serialização	2	20	20	512
BirdChicken	Serialização	2	20	20	512
Car	Sensor	4	60	60	577
CBF	Sintético	3	30	900	128
ChlorineConcentration	Sintético	3	467	3.840	166
CinCECGtorso	ECG	4	40	1.380	1.639
Coffee	Espectrograma	2	28	28	286
Computers	Sensor	2	250	250	720

CricketX	Movimento	12	390	390	300
CricketY	Movimento	12	390	390	300
CricketZ	Movimento	12	390	390	300
DiatomSizeReduction	Serialização	4	16	306	345
DistalPhalanxOutlineAgeGroup	Serialização	3	139	400	80
DistalPhalanxOutlineCorrect	Serialização	2	276	600	80
DistalPhalanxTW	Serialização	6	139	400	80
Earthquakes	Sensor	2	139	322	512
ECG200	ECG	2	100	100	96
ECG5000	ECG	5	500	4.500	140
ECGFiveDays	ECG	2	23	861	136
ElectricDevices	Sensor	7	8.926	7.711	96
FaceAll	Serialização	14	560	1.690	131
FaceFour	Serialização	4	24	88	350
FacesUCR	Serialização	14	200	2050	131
Fish	Serialização	7	175	175	463
FordA	Sensor	2	1.320	3.601	500
FordB	Sensor	2	810	3.636	500
GunPoint	Movimento	2	50	150	150
Ham	Espectrograma	2	109	105	431
HandOutlines	Serialização	2	370	1.000	2.709
Haptics	Movimento	5	155	308	1.092
Herring	Serialização	2	64	64	512
InlineSkate	Movimento	7	100	550	1.882
InsectWingbeatSound	Onda	11	220	1.980	256
ItalyPowerDemand	Sensor	2	67	1.029	24
LargeKitchenAppliances	Sensor	3	375	375	720
Lightning2	Sensor	2	60	61	637
Lightning7	Sensor	7	70	73	319
MALLAT	Sintético	8	55	2.345	1.024
Meat	Espectrograma	3	60	60	448
MedicalImages	Serialização	10	381	760	99
MiddlePhalanxOutlineAgeGroup	Serialização	3	154	400	80
MiddlePhalanxOutlineCorrect	Serialização	2	291	600	80
MiddlePhalanxTW	Serialização	6	154	399	80
MoteStrain	Sensor	2	20	1.252	84
NonInvasiveFetalECG1	ECG	42	1.800	1.965	750
NonInvasiveFetalECG2	ECG	42	1.800	1.965	750
OliveOil	Espectrograma	4	30	30	570
OSULeaf	Serialização	6	200	242	427
PhalangesOutlinesCorrect	Serialização	2	1.800	858	80
Phoneme	Onda	39	214	1.896	1.024
Plane	Sensor	7	105	105	144
ProximalPhalanxOutlineAgeGroup	Serialização	3	400	205	80
ProximalPhalanxOutlineCorrect	Serialização	2	600	291	80
ProximalPhalanxTW	Serialização	6	205	400	80
RefrigerationDevices	Sensor	3	375	375	720

ScreenType	Sensor	3	375	375	720
ShapeletSim	Sintético	2	20	180	500
ShapesAll	Sintético	60	600	600	512
SmallKitchenAppliances	Sensor	3	375	375	720
SonyAIBORobotSurface	Sensor	2	20	601	70
SonyAIBORobotSurfaceII	Sensor	2	27	953	65
StarLightCurves	Onda	3	1.000	8.236	1.024
Strawberry	Espectrograma	2	370	613	235
SwedishLeaf	Serialização	15	500	625	128
Symbols	Serialização	6	25	995	398
SyntheticControl	Sintético	6	300	300	60
ToeSegmentation1	Serialização	2	40	228	277
ToeSegmentation2	Serialização	2	36	130	343
Trace	Sensor	4	100	100	275
TwoLeadECG	ECG	2	23	1139	82
TwoPatterns	Sintético	4	1.000	4.000	128
uWaveGestureLibraryAll	Movimento	8	896	3.582	945
uWaveGestureLibraryX	Movimento	8	896	3.582	315
uWaveGestureLibraryY	Movimento	8	896	3.582	315
uWaveGestureLibraryZ	Movimento	8	896	3.582	315
Wafer	Sensor	2	1.000	6174	152
Wine	Espectrograma	2	57	54	234
WordsSynonyms	Serialização	25	267	638	270
Worms	Movimento	5	77	181	900
WormsTwoClass	Movimento	2	77	181	900
Yoga	Serialização	2	300	3.000	426

Os conjuntos de dados listados na Tabela 1 podem ser obtidos no endereço *online* em (CHEN *et al.*, 2015). As informações numéricas mostradas na tabela são obtidas dos próprios dados, porém a informação de domínio de aplicação não está disponível no repositório. O repositório lista os autores dos conjuntos, mas (com poucas exceções) não referencia os trabalhos que descrevem como os dados foram coletados ou compilados, portanto não foi possível incluir nesta Tese informações detalhadas em todos os casos. Por exemplo, os dados do conjunto Wafer foram obtidos de diversos sensores no processo de confecção de *chips* eletrônicos. O nome do conjunto advém do passo inicial desse processo: a confecção de um “bloco” de silício chamado “*wafer*”. Na Tese de (OLSZEWSKI, 2001) o autor explica em detalhes o processo de confecção de *chips* e quais sensores foram utilizados para coletar os dados temporais utilizados em seus experimentos. Entretanto, o repositório UCR não esclarece como os dados compilados pelo autor original foram selecionados e particionados nos exemplos de treinamento e teste, nem quais sensores foram escolhidos, para compor o conjunto de dados que foi finalmente disponibilizado pelos mantenedores.

Em que pese a ausência de informação sobre todos os dados, são notórias a diversidade do repositório e sua prevalência como *benchmark* em trabalhos acadêmicos de classificação de

séries temporais por similaridade—veja, por exemplo, (GENOLINI *et al.*, 2016; KATE, 2016; WANG; WANG; LIU, 2016; FAHIMAN *et al.*, 2016)¹³. Os conjuntos listados na Tabela 1 podem ser distribuídos nas seguintes categorias:

ECG: sequências de eletrocardiogramas. Dados de ECG são obtidos de um ou mais sensores elétricos (eletrodos) normalmente posicionados sobre a pele de um paciente. Um exame típico pode empregar até doze eletrodos e produzir centenas ou milhares de observações, mas os conjuntos de ECG no repositório UCR são dados segmentados de apenas um eletrodo ou alguma combinação linear de dois ou mais eletrodos. O conjunto ECG200, por exemplo, foi compilado por Olszewski (2001) a partir de um exame realizado com um único paciente durante um período de meia hora. Para produzir as séries temporais do repositório, o autor segmentou aleatoriamente os dados de dois eletrodos selecionando trechos nos quais se verificou a ocorrência de dois tipos de anomalias;

Espectrograma: dados obtidos de espectroscopia. Existem diferentes técnicas de espectroscopia que, de modo geral, consistem em utilizar ondas eletromagnéticas para estudar a composição e a propriedade de algum composto (NAWROCKA; LAMORSKA, 2013). Por exemplo, um alimento pode ser cozido e isso libera vapor ou fumaça. O espectro de absorções de um feixe de luz que cruza esse vapor ou fumaça é indicativo dos compostos presentes no alimento;

Movimento: dados que representam ou resumizam trajetórias. Os conjuntos CricketX/Y/Z registram cada qual uma única dimensão de dados de acelerômetros. O mesmo é observado também nos conjuntos uWaveGestureLibrary, porém no conjunto uWaveGestureLibraryAll os dados das três dimensões foram concatenados em uma única série. Já o conjunto GunPoint foi coletado de imagens: um ator foi filmado em duas situações. Na coleta das séries da classe “Gun”, o ator saca uma réplica de arma de um coldre preso à sua cintura, mira um ponto fixo diante de si por aproximadamente um segundo e devolve a arma ao coldre. As observações são a distância da mão do ator com relação a um ponto fixo. Já na coleta das séries da classe “NoGun”, o ator realiza o mesmo movimento, porém com a mão nua. O objetivo do problema é identificar os casos nos quais o ator sacou a arma e nos quais ele simulou o processo;

Onda: conjuntos de dados coletados diretamente de ondas mecânicas ou eletromagnéticas (exceto espectrogramas). Inclui exemplos de áudio (*e.g.*, InsectWingbeatSound e Phoneme) e curvas de luzes (StarLightCurves);

¹³ Foram selecionados quatro trabalhos recentes de autores que não estiveram diretamente envolvidos na criação ou manutenção do repositório UCR e que não foram, de acordo com o Google Scholar (<scholar.google.com>), co-autores de pesquisadores diretamente envolvidos na criação ou manutenção do repositório UCR em artigos publicados nos últimos cinco anos. Sem discriminação de autores, o número de artigos que referenciam o repositório UCR, de acordo com o Google Scholar, é superior a 350.

Serialização: tratam-se de conjuntos de dados nos quais as séries foram obtidas através do processo de serialização discutido na Seção 2.2. Diversos conjuntos dessa classe (*e.g.*, DistalPhalanxTW) resultaram de estudos de análise de imagens nos quais os pesquisadores tinham como interesse identificar o contorno da mão em imagens de radiografia (DAVIS *et al.*, 2011);

Sensor: dados reais coletados de procedimentos que não se encaixam nas demais categorias. Por exemplo, os conjuntos FordA e FordB são dados de diferentes sensores instalados em um veículo. Já o conjunto Wafer (NAWROCKA; LAMORSKA, 2013) contém dados de diferentes sensores utilizados no processo de fabricação de *chips* eletrônicos. O conjunto Earthquakes contém séries temporais coletadas de leituras de sismógrafos;

Sintético: todos os conjuntos de dados produzidos artificialmente são agrupados como “sintéticos”. Provavelmente o mais conhecido é o CBF, no qual as classes são variações de três séries temporais prototípicas, mostradas na Figura 13. Conjuntos sintéticos são interessantes porque permitem produzir grandes quantidades de exemplos rotulados com baixo custo e permitem a realização de experimentos diversos.

Figura 13 – Séries prototípicas do conjunto CBF.

O repositório UCR é visivelmente rico. Além de englobar vários domínios de aplicação, o repositório é diverso em termos de características do problema de classificação. Vários são problemas de classificação binária, porém 42 conjuntos possuem três ou mais classes e 15 possuem dez ou mais classes. O maior conjunto em número de classes é o ShapesAll, que possui 60 classes, cada qual representada por dez exemplos de treinamento—trata-se de um conjunto sintético de serialização. Já o conjunto 50Words é um caso de desbalanceamento extremo: 9 das 50 classes correspondem a 50% dos exemplos de treinamento e 21 classes são representadas por cinco instâncias de treinamento ou menos. Os conjuntos também variam bastante em número de observações—o menor possui apenas 24 observações e o maior, 2.709; a média é 422,21 observações e a mediana, 300.

Todos os conjuntos foram *z*-normalizados pelos mantenedores do repositório. Além disso, todos os conjuntos foram redimensionados para que as séries tenham o mesmo número de

observações. Em alguns casos, isso foi feito por meio de segmentação cuidadosa—no conjunto GunPoint, os autores utilizaram um metrônomo para que as séries fossem coletadas de forma homogênea. Em outros casos, os mantenedores do repositório utilizaram-se de interpolação ou sub-amostragem para reduzir a dimensionalidade dos dados. É o caso, por exemplo, do conjunto de dados Phoneme. Os dados originais (HAMOONI; MUEEN, 2014) consistem de aproximadamente 370 mil séries com tamanhos que variam de algumas dezenas a aproximadamente 5 mil observações. Os mantenedores do repositório UCR selecionaram um subconjunto de apenas 214 exemplos de treinamento redimensionados para 1.024 observações (algumas interpoladas para um número de observações maior que o original; outras sub-amostradas). Como ficará evidente ao longo desta Tese, trata-se do conjunto mais “difícil” do repositório.

Em alguns experimentos, foram utilizados conjuntos de dados que não pertencem ao repositório UCR. Para a parte principal deste projeto (*i.e.*, a comparação de diferentes representações e funções de distância), entretanto, os experimentos foram conduzidos principalmente com os conjuntos listados na Tabela 1. Os demais conjuntos serão apresentados nos capítulos convenientes.

2.6.2 Procedimentos de normalização

Embora a z -normalização seja provavelmente a técnica de normalização mais empregada na classificação de séries temporais por similaridade, existem outras estratégias de normalização que podem ser eficientes em alguns casos. Um destes, como será mostrado no Capítulo 4, é quando são utilizadas funções de distância que exigem propriedades específicas das séries. Porém mesmo quando se utiliza a distância euclidiana, existem situações nas quais normalizar para o intervalo $[0, 1]$, normalizar como histogramas ou não normalizar pode levar a comparações mais robustas entre as séries.

Na Seção 3.9.3, serão mostrados experimentos nos quais essas três técnicas de normalização foram comparadas sobre os dados do repositório UCR. Entretanto, como esses conjuntos são distribuídos previamente z -normalizados, não é possível utilizá-los para demonstrar a diferença entre z -normalização contra o caso no qual nenhuma técnica de normalização é empregada. Para esse tipo de experimento, foram utilizados os seguintes conjuntos de dados:

- Versões antigas dos conjuntos Beef, Coffee, CricketX, CricketY, CricketZ, Fish e OliveOil, obtidas do repositório UCR antes de sua mais recente atualização em 2015. Até essa época, o repositório UCR possuía 45 conjuntos de dados e os sete conjuntos ora mencionados não estavam z -normalizados;
- Versões originais dos conjuntos BeetleFly e BirdChicken. Originalmente, esses conjuntos foram disponibilizados *online* sem nenhum tipo de normalização. Tratam-se de conjuntos de dados temporais obtidos de serialização de imagens de insetos e aves (respectivamente);

- Os conjuntos Arcene¹⁴ e HAR¹⁵ (ANGUITA *et al.*, 2013) do *UCI Machine Learning Repository* (LICHMAN, 2013). O repositório UCI, hospedado pela universidade da Califórnia, campus de Irvine, contém (no momento em que esta Tese foi escrita) 360 conjuntos de dados destinados a várias tarefas de aprendizado de máquina. O conjunto Arcene possui 900 séries temporais com 1.000 observações obtidas de exames médicos relacionados a detecção de câncer. O conjunto HAR (*Human Activity Recognition Using Smartphones Data Set*) possui 10.299 instâncias de dados temporais nos quais as séries foram coletadas por dados de acelerômetros utilizados por indivíduos na realização de atividades do dia-a-dia.

Na Tabela 2 são apresentados os valores de acurácia obtidos para o classificador 1-NN, com distância euclidiana, nesses 11 conjuntos de dados temporais. O classificador 1-NN e o conceito de acurácia serão apresentados formalmente no próximo capítulo. Por ora, convém apenas que o classificador 1-NN depende fortemente do conceito de distância entre os objetos classificados. Portanto os escores apresentados na Tabela 2 refletem aproximadamente o grau de confiança que podemos depositar na função de distância.

Tabela 2 – Comparação de estratégias de normalização. Os resultados mostrados na tabela provêm de um experimento preliminar.

Conjunto	Tipo de normalização			
	Nenhum	Z-normalização	Intervalo [1,0]	Histograma
Arcene	0,8800	0,8800	0,8700	0,8600
Beef	0,5333	0,6667	0,6333	0,6667
BeetleFly	0,6500	0,7500	0,8000	0,8500
BirdChicken	0,6000	0,5500	0,6500	0,7000
Coffee	0,7500	1,0000	1,0000	0,9643
CricketX	0,5744	0,5769	0,5846	0,5436
CricketY	0,6436	0,5667	0,5513	0,5513
CricketZ	0,6205	0,5872	0,5923	0,5667
Fish	0,7829	0,7829	0,7771	0,7829
HAR	0,8785	0,8809	0,8775	0,8812
OliveOil	0,8667	0,8333	0,8333	0,8667

O objetivo desse experimento foi verificar se procede a percepção geral de que a z-normalização é uma “etapa fundamental” na classificação de séries temporais e se toda normalização dos dados deveria ser feita com z-normalização. Note que se trata de um experimento preliminar, utilizando um número bastante reduzido de conjuntos de dados.

Na maioria dos casos, foi possível encontrar um método de normalização que produziu escores compatíveis com ou superiores aos escores obtidos na ausência de normalização. Isso sugere que algum tipo de normalização dos dados pode ser benéfica para a etapa de classificação, embora não permita afirmar que seja possível definir *qual* será a melhor estratégia de normalização para um conjunto de dados qualquer.

¹⁴ <<http://archive.ics.uci.edu/ml/datasets/Arcene>>

¹⁵ <<http://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones>>

Outro resultado observado na tabela é que, em vários casos, a normalização no intervalo $[0, 1]$ ou como histogramas produziu escores tão bons ou melhores que a z -normalização. Embora essas observações não sejam apresentadas com significância estatística, foram motivadoras para o estudo que será apresentado na Seção 3.9.3 desta Tese.

2.7 Considerações finais

Séries temporais são um importante tipo de dados que encontram aplicações em diversas áreas do conhecimento humano. Muitas dessas aplicações dependem de um conceito de similaridade capaz de indicar, por exemplo, qual é o par de séries mais semelhantes em um conjunto de dados temporais. Esse conceito é tipicamente expresso como uma função de distância, a qual pode ser ocasionalmente associada a um ou mais tipos de invariância, como invariância a *warping* local ou a complexidade.

Outro importante tipo de invariância é a invariância a amplitude e *offset*. Esse tipo de invariância pode ser facilmente obtido através da normalização da série. Foram considerados três métodos de normalização: a z -normalização ou normalização em escores padrão, a normalização dos dados para o intervalo $[0, 1]$ e a normalização em histogramas. A normalização em escores padrão é certamente a mais comum nos trabalhos de classificação por similaridade que se aproximam desta Tese. Os outros dois métodos, embora não sejam tão comuns, foram utilizados neste trabalho com funções de distância específicas, como será mostrado no Capítulo 4.

Neste capítulo foi também apresentado o repositório UCR, oficialmente denominado “*UCR Time Series Classification Archive*”. Trata-se, provavelmente, do maior repositório aberto de séries rotuladas. Seu uso é prevalente na literatura científica relacionada a classificação por similaridade. No restante desta Tese, serão levantadas várias hipóteses, as quais serão verificadas por meio de experimentos no repositório UCR.

APRENDIZADO DE MÁQUINA

3.1 Considerações iniciais

DE ACORDO COM (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996), **extração de conhecimento de bases de dados** (KDD, do inglês *Knowledge Discovery in Databases*) é “o processo não trivial de identificação de padrões válidos, novos, potencialmente úteis e compreensíveis nos dados”. Um registro de todas as transações realizadas em um mercado, por exemplo, é um conjunto de dados. Empregar KDD nesse tipo de dado consiste em selecionar e processar os dados, identificar padrões e interpretá-los, gerando conhecimento sobre o comportamento dos clientes, dos trabalhadores, da popularidade dos produtos etc. Ainda de acordo com (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996), **mineração de dados** (MD) é a aplicação de algoritmos específicos para identificar padrões nos dados.

Aprendizado de máquina (AM) é uma linha de pesquisa em inteligência artificial que tem por objetivos a pesquisa de técnicas relacionadas ao aprendizado e a construção de sistemas capazes de adquirir conhecimento de forma automática. O aprendizado de máquina é efetivamente empregado na mineração de dados para construir modelos capazes de identificar padrões automaticamente, utilizando para isso métodos estatísticos que se baseiam em inferências sobre uma amostra dos dados (ALPAYDIN, 2010).

Um exemplo bastante relevante de aplicação de AM na atualidade é o desenvolvimento de sistemas capazes de analisar risco de crédito financeiro (KHANDANI; KIM; LO, 2010). Esse tipo de sistema é utilizado por agências concessionárias de crédito para prever se um potencial cliente irá conseguir saldar ou não uma dívida. Nesse caso os dados são registros de empréstimos passados, juntamente com as conclusões desses casos. O conhecimento desejado pode ser, por exemplo, um conjunto de critérios para prever a conclusão de um caso futuro.

Para que esse problema possa ser resolvido por meio de uma abordagem de AM, é necessário primeiramente que os dados sejam representados como exemplos rotulados. A informação

de cada caso é armazenada como um exemplo $E = (x, y)$, no qual $x = (x_1, x_2, \dots, x_M)$ é um conjunto de valores relacionados a um conjunto de atributos $X = (X_1, X_2, \dots, X_M)$. É importante que os atributos sejam adequados para descrever os exemplos de acordo com o contexto no qual se emprega KDD. Por exemplo, *casado* = {sim, não} é provavelmente um atributo relevante para análise de risco de crédito, pois expressa informação relacionada a diversos indicadores sócio-econômicos que podem ser necessários para caracterizar o quadro financeiro de um potencial cliente. Por outro lado, o atributo *altura* $\in \mathbb{R}$ provavelmente é irrelevante nesse contexto. Já o valor $y \in Y$ diz respeito a um atributo especial denominado **classe**.

A presença ou a ausência da classe e seus possíveis valores são características cruciais em um problema de mineração de dados. O objetivo da classe é expressar o conceito que se deseja associar aos exemplos. Um caso simples para risco de crédito poderia ser o espaço $Y = \{\text{pagou, não pagou}\}$. Nesse caso o conhecimento adquirido apenas poderia indicar se um potencial cliente irá pagar ou não o empréstimo. Uma alternativa provavelmente mais interessante para profissionais da área seria um espaço de classes $Y = \{\text{baixo, médio, alto}\}$, indicando diferentes graus de risco de crédito. A escolha deve levar em consideração tanto o propósito do conhecimento desejado quanto o custo necessário para obtê-lo—no primeiro caso, basta registrar se o pagamento foi efetuado; no segundo pode ser necessária uma avaliação *post-mortem* do risco corrido.

Consideremos um segundo exemplo. O termo “*spam*” é utilizado para designar todo tipo de *email* não desejado, sendo mais usualmente associado a *emails* que são enviados para um grande número de destinatários na tentativa de oferecer um produto ou serviço não solicitado. Estima-se que, entre janeiro e junho de 2014, aproximadamente 90% de todos os *emails* enviados eram *spam*¹. No passado, clientes de *email* tipicamente ofereciam aos usuários apenas a opção de criar filtros, baseados em regras manualmente especificadas, que poderiam ser utilizados para identificar automaticamente alguns tipos de *spam*. Atualmente, clientes de *email* empregam aprendizado de máquina para produzir filtros anti-*spam* de forma automática e contínua.

Ambos os exemplos dados nesta seção consideram problemas parecidos, nos quais existe um conceito a ser aprendido e o objetivo é associar exemplos a um conceito pré-determinado. A principal diferença entre cada caso é o tipo de dado envolvido. No primeiro, os dados são **estruturados** como valores de atributos. No segundo, os dados são sequências de palavras que não seguem uma estrutura comum. Esse tipo de dado é denominado **não estruturado** e requer técnicas distintas daquelas que são empregadas para lidar com dados estruturados. Grande esforço da comunidade de mineração de textos é dedicado a técnicas de pré-processamento para reduzir as dificuldades inerentes ao processamento de dados não estruturados (VIJAYARANI; ILAMATHI; NITHYA, 2015).

Séries temporais constituem um caso particular de dados não estruturados. Com respeito

¹ De acordo com relatório do *Messaging, Malware and Mobile Anti-Abuse Working Group*, disponível online no endereço eletrônico <www.m3aawg.org/for-the-industry/email-metrics-report>.

à *classificação de séries temporais*, as duas principais vertentes são a extração de atributos e a classificação por instâncias. Neste capítulo, são apresentados os conceitos fundamentais de aprendizado indutivo supervisionado e classificação de dados. Na Seção 3.5, apresenta-se uma revisão da literatura referente a classificação por atributos e por similaridade. Nas seções 3.6 e 3.7 são apresentadas estratégias para avaliar classificadores e algoritmos de indução de classificadores. Na Seção 3.8, aborda-se o uso das técnicas de avaliação de classificadores para a seleção de parâmetros de algoritmos de aprendizado de máquina. Finalmente, na Seção 3.9, são apresentados resultados experimentais referentes aos tópicos abordados neste capítulo.

3.2 Classificação supervisionada

O objetivo da **classificação** é examinar um objeto e decidir, com respeito a um conjunto discreto de classes ou rótulos conhecidos, a qual classe ele pertence (RICH; KNIGHT, 1991). Decidir se um *email* deve ser considerado *spam*, por exemplo, é um problema de classificação. Um algoritmo que realiza essa decisão é denominado **classificador**.

Classificadores podem ser construídos utilizando-se aprendizado indutivo supervisionado. O aprendizado é dito ser **indutivo** porque é realizado por um algoritmo que induz ou extrapola, a partir de um conjunto limitado de dados, o conceito que permite realizar a classificação. O aprendizado é **supervisionado** quando os dados são rotulados; isto é, cada objeto apresentado ao algoritmo indutor é acompanhado de um rótulo.

Mais formalmente, os objetos podem ser representados como pares (x, y) , no qual $x \in X$ é um vetor de valores que representa uma **instância** dos objetos a serem classificados. O rótulo $y \in Y$ representa o conceito associado à instância e assume-se que existe uma função $\zeta : X \mapsto Y$, denominada **função-conceito**, que relaciona qualquer instância x_i ao seu rótulo $\zeta(x_i) = y_i$. Um **conjunto de treinamento** é um conjunto de exemplos rotulados $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$.

O objetivo ideal é **induzir** a função-conceito ζ a partir do conjunto de treinamento D . Entretanto, esse objetivo é em geral inalcançável, pois as amostras são tipicamente insuficientes para caracterizar perfeitamente a função-conceito. Assim, o objetivo prático é encontrar, com base nos exemplos de treinamento D , uma boa aproximação $\hat{h}$ tal que $\hat{h} \approx \zeta$ (RUSSELL; NORVIG, 2010). Nesse caso o mapeamento $\hat{h} : X \mapsto Y$ é uma **hipótese** ou um **modelo** sobre ζ .

A premissa fundamental do aprendizado indutivo sustenta que qualquer modelo ou hipótese que aproxime bem a função-conceito para um conjunto de treinamento representativo também será capaz de aproximar bem a função-conceito para exemplos não observados (MITCHELL, 1997). Porém, o espaço de hipóteses sobre ζ é potencialmente infinito e pode conter inúmeros modelos que aparentem ser boas aproximações de ζ nas amostras, mas não generalizem bem para exemplos novos. É necessário que o indutor estabeleça critérios para restringir o processo de busca a um espaço de hipóteses factível H e “critérios de preferência” que lhe

permitam escolher uma hipótese adequada nesse espaço, em detrimento de outras hipóteses candidatas (HAUSSLER, 1988). Esse conjunto de critérios configura o **viés** do indutor.

3.3 Linguagem de descrição dos dados

Para que seja possível induzir um classificador, é necessário estabelecer uma linguagem formal de descrição dos dados. Um caso comum em AM ocorre quando os dados podem ser estruturados e representados na forma de uma tabela atributo-valor. A **tabela atributo-valor** é uma linguagem que possui poder de descrição equivalente ao da lógica proposicional. Nessa linguagem cada exemplo é descrito como um par $E_i = (x_i, y_i)$ no qual $x = (x_1, x_2, \dots, x_M)$ é um conjunto de valores ou coeficientes relacionados a um conjunto de atributos $X_1, X_2, \dots, X_M$ e y_i é um atributo denominado classe que assume uma das classes conhecidas $\{C_1, C_2, \dots, C_k\}$. Os valores C_j do espaço de classes Y são também denominados rótulos.

No restante desta Tese, seguiremos a notação empregada por Mitchell (1997) na qual o conjunto de valores $x_i \in X$ é uma **instância** do conceito a ser aprendido. No caso específico em que os dados são estruturados, o espaço no qual as instâncias residem é um espaço de atributos. Cada atributo representa um aspecto ou característica dos objetos. Um atributo, isoladamente, não é necessariamente capaz de distinguir os objetos, mas todos os objetos que possuem o mesmo valor para algum atributo devem compartilhar alguma característica.

Em algumas aplicações, os atributos podem assumir valores desconhecidos ou ausentes. Valores desconhecidos ou ausentes podem surgir por diversos motivos (BATISTA; MONARD, 2003; LAKSHMINARAYAN; HARP; SAMAD, 1999; ZHENG; LOW, 1999), incluindo falhas ou restrições práticas no processo de amostragem dos dados. Por exemplo, existem casos nos quais a amostragem de algum atributo específico pode ser dispendiosa, de modo que essa informação não estará disponível para todos os exemplos. Em particular, quando a classe pode assumir valores desconhecidos, o problema é um caso de **aprendizado semissupervisionado**². Nesta Tese, por restrição de escopo, consideraremos apenas problemas nos quais os objetos são completamente caracterizados para todos os atributos.

Os atributos podem ser categorizados de acordo com propriedades dos valores que eles podem assumir. Por exemplo, os valores podem ser discretos ou contínuos, ordinais ou nominais etc. Em alguns casos, essa categorização deve ser feita com maior ou menor granularidade. Nesta Tese, consideram-se apenas atributos quantitativos e qualitativos. Os **atributos quantitativos** são aqueles que representam uma quantidade numérica escalar ou unitária, como Altura, Idade e Temperatura.

Já os atributos **qualitativos** são considerados aqueles cujos possíveis valores pertencem a

² Note que a distinção entre valores ausentes e desconhecidos é importante para o atributo classe: um novo exemplo a ser classificado possui classe desconhecida, mas presente. Em contraste, no agrupamento não existem classes associadas às instâncias—*i.e.*, elas são ausentes.

um conjunto discreto, finito e não ordinais, como Gênero (musical, de sexo etc.). Alguns atributos que assumem valores tipicamente ordinais podem ser tratados como qualitativos. Por exemplo, o atributo Altura pode ser tratado como qualitativo se for discretizado em faixas (e.g., “baixa”, “média”, “alta”) e a ordem entre os valores for irrelevante para o problema de classificação.

A classe também pode ser quantitativa ou qualitativa. No caso em que ela assume valores quantitativos, o problema é especificamente denominado **regressão**. Problemas de classificação nos quais a classe pode assumir apenas um dentre dois possíveis valores são denominados problemas de **classificação binária**. Todos os problemas de classificação binária são equivalentes ao problema no qual o espaço de classes é $Y = \{\text{positivo}, \text{negativo}\}$ ou $Y = \{\oplus, \ominus\}$. Quando o conjunto de valores possíveis para a classe possui mais que dois valores, denomina-se **multiclasse**. Problemas de classificação multiclasse são distintos de problemas de classificação *multirrótulo*, nos quais a classe é um vetor de um ou mais valores.

Como um exemplo que será utilizado no restante deste capítulo, apresenta-se na Tabela 3 uma tabela atributo-valor que representa o conjunto de dados tênis. Cada exemplo desse conjunto é um registro das condições meteorológicas do dia em que um indivíduo optou por ir ou não jogar tênis. O objetivo da classificação é encontrar os critérios utilizados pelo indivíduo para tomar a decisão.

Tabela 3 – Conjunto de dados tênis representado como tabela atributo-valor. Exemplo adaptado de (QUINLAN, 2014).

	Aspecto	Temperatura	Umidade	Vento	JogarTênis
E ₁	Aberto	Alta	Elevada	Fraco	NãoJogar
E ₂	Aberto	Alta	Elevada	Forte	NãoJogar
E ₃	Nublado	Alta	Elevada	Fraco	Jogar
E ₄	Chuvoso	Média	Elevada	Fraco	Jogar
E ₅	Chuvoso	Baixa	Normal	Fraco	Jogar
E ₆	Chuvoso	Baixa	Normal	Forte	NãoJogar
E ₇	Nublado	Baixa	Normal	Forte	Jogar
E ₈	Aberto	Média	Elevada	Fraco	NãoJogar
E ₉	Aberto	Baixa	Normal	Fraco	Jogar
E ₁₀	Chuvoso	Média	Normal	Fraco	Jogar
E ₁₁	Aberto	Média	Normal	Forte	Jogar
E ₁₂	Nublado	Média	Elevada	Forte	Jogar
E ₁₃	Nublado	Alta	Normal	Fraco	Jogar
E ₁₄	Chuvoso	Média	Elevada	Forte	NãoJogar

Todos os atributos do conjunto tênis são qualitativos. O atributo Temperatura, do qual normalmente se esperaria uma representação na escala Celsius ou na escala Fahrenheit, foi **discretizado** para os valores Baixo, Médio e Alto. Essa adaptação dos dados originais de (QUINLAN, 2014), no qual a temperatura foi originalmente dada em °F, simplifica a apresentação dos indutores na Seção 3.4.

3.4 Linguagem de descrição do modelo

Uma colocação que pode ser considerada trivial, mas de fundamental importância para o aprendizado de máquina, é que conhecimento é um conceito abstrato. Não se pode escrever, desenhar ou tabular o conhecimento, apenas representá-lo através de um modelo que pode ser um texto, uma figura ou uma tabela. Ou seja, o modelo é um “substituto” do conhecimento. Por esse motivo, é necessário estabelecer um formalismo de descrição para representá-lo.

A linguagem de representação do modelo é o viés de indução mais prevalente na mineração de dados. Quando escolhemos uma linguagem, estamos efetivamente restringindo o conjunto de hipóteses sobre a função-conceito àquelas podem ser representadas por meio dessa linguagem. Por exemplo, árvores de decisão tradicionais estabelecem relações entre atributos e seus possíveis valores, não sendo capazes de representar relações entre atributos. Quando escolhemos induzir árvores de classificação, estamos automaticamente descartando todas as hipóteses que requerem esse tipo de relação—*e.g.*, altura > largura.

Linguagens de descrição podem ser simbólicas ou não simbólicas. A distinção entre os dois casos é intuitiva: linguagens e modelos destinados à interpretação humana são denominados **simbólicos**. Linguagens e modelos destinados a processamento automático são denominados **não simbólicos**. Exemplos de modelos simbólicos incluem a **regra de conhecimento** ou de produção, que pode descrever classificadores segundo a forma geral “se <condições> então <classe = C_j >”. Exemplos de modelos não simbólicos incluem as **máquinas de vetores de suporte**, que descrevem conhecimento como um mapeamento do espaço de instâncias para um espaço hiperdimensional implícito (Seção 5.7).

3.4.1 Modelos probabilísticos de Bayes

Modelos probabilísticos de Bayes são diversos modelos (normalmente considerados não simbólicos) que estabelecem inferências utilizando a relação de Bayes. A relação de Bayes (Equação 3.1) é um importante resultado da estatística Bayesiana que infere a probabilidade de ocorrência do evento A dado que se tem a informação prévia da ocorrência do evento B .

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (3.1)$$

Em (3.1), $P(A)$ e $P(B)$ são denominadas probabilidades a priori dos eventos. Seus valores refletem a informação que podemos associar aos eventos antes que eles sejam observados. $P(A|B)$ e $P(B|A)$ são denominadas probabilidades a posteriori e seus valores refletem a informação que podemos associar a um evento quando temos a informação adicional de que o outro evento foi observado.

A relação de Bayes pode ser utilizada para construir um classificador de Bayes. Considere um conjunto de hipóteses H sobre uma função-conceito ζ , um conjunto de dados D e um espaço

de classes $Y = \{C_1, C_2, \dots, C_k\}$. As seguintes probabilidades a priori e a posteriori são necessárias para a definição do modelo:

- $P(h_i)$ é a probabilidade a priori de que a hipótese h_i é verdadeira, desprovido de qualquer conhecimento sobre a distribuição dos dados ou das classes;
- $P(C_j|h_i)$ é a probabilidade a posteriori de que um exemplo será classificado como C_j dado que a hipótese h_i é verdadeira;
- $P(h_i|D)$ é a probabilidade a posteriori de que a hipótese h_i é verdadeira dado que os exemplos D foram observados.

Com base nessas relações, o **classificador ótimo de Bayes** é aquele que classifica uma nova instância x como a classe C_j que maximiza a probabilidade $P(C_j|D)$,

$$y_i = \operatorname{argmax}_{C_j \in Y} \sum_{h_i \in H} P(C_j|h_i) P(h_i|D). \quad (3.2)$$

A probabilidade a posteriori $P(h_i|D)$ reflete a chance de que a hipótese h_i seja verdadeira em um cenário no qual os dados D são observáveis. O classificador pondera essa probabilidade pela chance de que a classe verdadeira do exemplo seja *realmente* C_j em um cenário no qual a hipótese h_i é assumida como verdade. Em outras palavras, o classificador ótimo de Bayes considera *todas* as possíveis hipóteses que podem ser utilizadas para aproximar um conceito e “aposta” na hipótese mais provável para os dados observados.

Considere o seguinte exemplo de classificação usando o modelo ótimo de Bayes—adaptado de (MITCHELL, 1997).

Suponha um problema de classificação binária para o qual existe uma função-conceito desconhecida $\zeta : X \mapsto \{\oplus, \ominus\}$. Suponha que o espaço de hipóteses admite apenas as hipóteses h_1 , h_2 , e h_3 e que as suas probabilidades a posteriori, dado um conjunto de exemplos D , são respectivamente 0,4; 0,3 e 0,3.

Seja x uma nova instância, a qual foi classificada como positiva por h_1 , mas negativa por h_2 e h_3 . Assim, as probabilidades a posteriori das classes, dadas as hipóteses, são as seguintes.

$$\begin{aligned} P(\oplus|h_1) &= 0 & P(\ominus|h_1) &= 1 \\ P(\oplus|h_2) &= 1 & P(\ominus|h_2) &= 0 \\ P(\oplus|h_3) &= 1 & P(\ominus|h_3) &= 0 \end{aligned}$$

Portanto,

$$\begin{aligned} \sum_{h_i \in H} P(\oplus|h_i) P(h_i|D) &= P(\oplus|h_1) P(h_1|D) &= 0,4, \\ \sum_{h_i \in H} P(\ominus|h_i) P(h_i|D) &= P(\ominus|h_2) P(h_2|D) + P(\ominus|h_3) P(h_3|D) &= 0,6 \end{aligned}$$

e

$$\operatorname{argmax}_{C_j \in \{\oplus, \ominus\}} \sum_{h_i \in H} P(C_j|h_i) P(h_i|D) = \ominus.$$

De modo que o classificador ótimo de Bayes irá classificar a nova instância como sendo da classe negativa.

O classificador ótimo de Bayes é o melhor modelo possível no caso médio (RUSSELL; NORVIG, 2010). A taxa de acerto desse classificador é um limite superior de qualquer modelo que pode ser induzido sobre um conjunto de dados D . Entretanto, esse classificador é geralmente impraticável, pois o espaço de hipóteses H pode ser muito grande ou mesmo infinito. Além disso, encontrar ou mesmo estimar as probabilidades posteriores necessárias para maximizar a relação (3.2) requer uma enumeração exponencial e tem custo computacional tipicamente proibitivo, mesmo para espaços de decisão consideravelmente modestos.

Podemos obter um modelo aproximado e consideravelmente mais simples assumindo que os atributos são independentes entre si. A probabilidade condicional conjunta da Equação 3.3 pode ser reescrita como a multiplicatória na Equação 3.4.

$$P(C_j|x) = \frac{P(x_1, x_2, \dots, x_M|C_j) \cdot P(C_j)}{P(x_1, x_2, \dots, x_M)} \quad (3.3)$$

$$P(C_j|x) = P(C_j) \cdot \prod_{i=1}^M \frac{P(x_i|C_j)}{P(x_i)} \quad (3.4)$$

Note que, uma vez observado o exemplo, as probabilidades $P(x)$ são constantes para cada classe. Isso nos permite descrever a seguinte relação de proporcionalidade.

$$P(C_j|x) \propto P(C_j) \cdot \prod_{i=1}^M P(x_i|C_j) \quad (3.5)$$

Consideremos ainda que as probabilidades envolvidas na relação (3.5) podem ser estimadas com base em uma pequena amostra dos dados. Assim, o **classificador ingênuo de Bayes**, mais usualmente conhecido como **Naive Bayes** (NB), é o modelo de classificação que estima a classe de um exemplo (x, y) encontrando um argumento que satisfaz a relação (3.6).

$$\hat{y} = \operatorname{argmax}_{j \in \{1, \dots, k\}} P(C_j) \prod_{i=1}^M P(x_i|C_j). \quad (3.6)$$

Considere como exemplo o conjunto de dados tênis, apresentado na Tabela 3. As probabilidades a priori das classes podem ser estimadas como o número de exemplos nos quais cada classe é observada. Ou seja, $P(\text{Jogar}) = \frac{9}{15}$ e $P(\text{NãoJogar}) = \frac{6}{15}$. As probabilidades posteriores dos atributos são estimadas com base na ocorrência dos valores dos atributos nos subconjuntos para os quais $y_i = \text{Jogar}$ e $y_i = \text{NãoJogar}$. Por simplicidade, apresenta-se a seguir apenas as

probabilidades a posteriori necessárias para classificar um único exemplo de teste, não presente no conjunto de treinamento, $E_{15} = ((\text{Chuvoso}, \text{Alta}, \text{Normal}, \text{Fraco}), \text{Jogar})$.

$$\begin{array}{ll} P(\text{Aspecto} = \text{Chuvoso} | \text{Jogar}) = \frac{3}{9} & P(\text{Aspecto} = \text{Chuvoso} | \text{NãoJogar}) = \frac{2}{5} \\ P(\text{Temperatura} = \text{Alta} | \text{Jogar}) = \frac{2}{9} & P(\text{Temperatura} = \text{Alta} | \text{NãoJogar}) = \frac{2}{5} \\ P(\text{Umidade} = \text{Normal} | \text{Jogar}) = \frac{6}{9} & P(\text{Umidade} = \text{Normal} | \text{NãoJogar}) = \frac{1}{5} \\ P(\text{Vento} = \text{Fraco} | \text{Jogar}) = \frac{6}{9} & P(\text{Vento} = \text{Fraco} | \text{NãoJogar}) = \frac{2}{5} \end{array}$$

As probabilidades de que a classe correta de E_{15} seja Jogar e NãoJogar são proporcionais a:

$$\begin{aligned} P(\text{Jogar} | E_{15}) &= \frac{9}{15} \cdot \frac{3}{9} \cdot \frac{2}{9} \cdot \frac{6}{9} \cdot \frac{6}{9} = 0,020 \\ P(\text{NãoJogar} | E_{15}) &= \frac{6}{15} \cdot \frac{2}{5} \cdot \frac{2}{5} \cdot \frac{1}{5} \cdot \frac{2}{5} = 0,005 \end{aligned}$$

Portanto, um classificador ingênuo de Bayes induzido sobre o conjunto de dados ténis seria capaz de classificar corretamente o exemplo E_{15} como pertencente à classe Jogar.

A linguagem de descrição para modelos NB é simplesmente uma tabela de probabilidades ou um conjunto de probabilidades a priori e a posteriori. Um algoritmo indutor que produz um conjunto de probabilidades é apresentado no Algoritmo 3.1.

Algoritmo 3.1: Indutor de modelo de classificação *Naive Bayes*

```

1 função Modelo = NaiveBayes (D, Atributos, Classes)
  Parâmetros de Entrada:
    D: conjunto de dados rotulados
    Atributos: descrição do espaço de atributos
    Classes: descrição do espaço de classes
  Parâmetros de Saída:
    Modelo: um conjunto de probabilidades a priori e a posteriori
2 para cada rótulo  $C_j$  do espaço de classes descrito por Classes faça
3    $D_j \leftarrow$  exemplos em D com classe  $y_i = C_j$ 
4    $P(C_j) \leftarrow \frac{|D_j|}{|D|}$ 
5   Registre a probabilidade a priori  $P(C_j)$  a Modelo
6   para cada Atributo  $\in$  Atributos faça
7     para cada possível valor  $x$  do atributo Atributo faça
8        $f_x \leftarrow$  número de exemplos em  $D_j$  para os quais Atributo =  $x$ 
9        $P(\text{Atributo} = x | C_j) \leftarrow \frac{f_x}{|D_j|}$ 
10      Registre a probabilidade a posteriori  $P(\text{Atributo} = x | C_j)$  em Modelo
11    fim
12  fim
13 fim

```

Uma restrição do indutor descrito pelo Algoritmo 3.1 é que as probabilidades são sempre calculadas com a suposição de que os atributos são qualitativos. Uma simples adaptação para atributos quantitativos poderia ser substituir os passos 8 e 9 do Algoritmo 3.1 para calcular a probabilidade a posteriori $P(x_i|C_j)$ de acordo com alguma distribuição de probabilidades contínua. Assim como as probabilidades posteriores são estimadas, para atributos qualitativos, dos subconjuntos referentes às classes D_j , os parâmetros dessa distribuição devem ser estimados dos valores para os exemplos de D_j . Por exemplo, pode-se assumir que a probabilidade a posteriori toma uma distribuição gaussiana $\mathcal{N}(m_j, s_j^2)$. Nesse caso m_j é a média dos valores do atributo para os exemplos de treinamento cujos rótulos são C_j e s_j é o desvio padrão amostrado dos mesmos exemplos. A probabilidade de que um exemplo assuma o valor $X_i = x_i$ para um atributo X_i , dada a classe C_j , é definida pela Equação 3.7.

$$P(x_i|C_j) = \frac{1}{\sqrt{2\pi s_j^2}} e^{-\frac{(x_i - m_j)^2}{2s_j^2}} \quad (3.7)$$

3.4.2 Modelos de árvore de decisão

A **árvore de decisão** é uma linguagem simbólica que representa as hipóteses como generalizações de um grafo acíclico não direcionado. Nesse modelo, cada vértice é associado a um atributo, podendo haver mais de um vértice associado ao mesmo atributo. As arestas estabelecem relações sobre os atributos, podendo restringi-los a um valor específico ou impor-lhes limites inferiores ou superiores. Os vértices terminais (folhas) são relacionados às classes. Um caminho da raiz até um vértice terminal estabelece condições sobre os atributos que equivalem a uma regra de conhecimento.

Por exemplo, considere novamente o conjunto de dados ténis e o exemplo de teste $E_{15} = ((\text{Chuvoso}, \text{Alta}, \text{Normal}, \text{Fraco}), \text{Jogar})$. Uma possível árvore de decisão para o conjunto ténis poderia ser como a mostrada na Figura 14. A classificação de qualquer exemplo, por meio de uma árvore de decisão, é um caminho condicionado aos valores do exemplo partindo da raiz em direção a algum vértice terminal. Para E_{15} , o caminho encontrado seria *raiz* → Aspecto é Chuvoso → Vento é Fraco → *vértice terminal* → *classificar como Jogar*.

O algoritmo mais conhecido para indução de árvores de decisão é, provavelmente, o C4.5 (QUINLAN, 2014). O C4.5 utiliza conceitos de teoria da informação para subdividir o espaço de decisão em sub-espacos mais homogêneos, nos quais as instâncias não observadas pertencem a alguma classe com maior probabilidade. A **entropia** ou, mais especificamente, a **entropia de Shannon**, é uma medida de “incerteza” ou “surpresa”. Um espaço de atributos que contém exemplos das classes Jogar e NãoJogar é mais “incerto” ou, equivalentemente, observar esse espaço causa mais “surpresa”, do que um espaço de atributos que contém apenas exemplos da classe Jogar. Como o espaço de atributos não é completamente conhecido, o indutor utiliza as amostras para estimar a entropia do espaço.

Figura 14 – Uma possível árvore de decisão para o conjunto de dados tênis apresentado na Tabela 3.

Definição 3.1: Seja $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$ um conjunto de exemplos rotulados. A entropia do atributo classe para os exemplos de D , dado que a classe pode assumir valores no espaço $Y = \{C_1, C_2, \dots, C_k\}$ é

$$H(D) = - \sum_{j=1}^k p(D, C_j) \cdot \log_2(p(D, C_j)), \quad (3.8)$$

na qual $p(D, C_j)$ é a frequência relativa do número de exemplos em D para os quais $y_i = C_j$:

$$p(D, C_j) = \frac{|\{(x_i, y_i) \in D \text{ tal que } y_i = C_j\}|}{|D|}. \quad (3.9)$$

O indutor C4.5 utiliza a entropia em um procedimento recursivo de divisão e conquista. A cada passo, o algoritmo considera diferentes particionamentos sobre D . Um **particionamento** de D é simplesmente uma coleção de subconjuntos não vazios de D que não se sobrepõem. Mais especificamente, um particionamento sobre D é dado pela Definição 3.2.

Definição 3.2: Seja $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$ um conjunto de exemplos. O conjunto $P_D = \{T_1, T_2, \dots, T_r\}$ é um particionamento sobre D se

1. $T_i \neq \emptyset$ para todo i ;
2. $T_1 \cup T_2 \cup \dots \cup T_r = D$; e
3. $T_i \cap T_j = \emptyset$ para todo $i \neq j$.

Em cada iteração, o C4.5 busca a estratégia de particionamento que produz menor entropia condicional sobre a classe. A entropia condicional é uma média das entropias das partições ponderada pelos tamanhos das partições.

Definição 3.3: Seja $P_D = \{T_1, T_2, \dots, T_r\}$ um particionamento sobre D . A entropia do particionamento é definida como a entropia condicional de D , dado o particionamento

$$H(D|P_D) \equiv \sum_{T_i \in P_D} P(T_i) \cdot H(D|T_i) = \sum_{i=1}^r \frac{|T_i|}{|D|} H(T_i). \quad (3.10)$$

Cada estratégia de particionamento considerada diz respeito aos valores de um atributo. O C4.5 escolhe o particionamento relacionado ao atributo de menor entropia e divide o conjunto D de acordo com esse particionamento. O algoritmo prossegue particionando os subconjuntos até que todos os exemplos pertençam a uma única classe.

Os critérios adotados pelo C4.5 para definir uma estratégia de particionamento variam de acordo com a natureza do atributo. Para atributos qualitativos, o particionamento é feito de modo que todos os exemplos de uma partição compartilham o mesmo valor para esse atributo e partições distintas assumem valores distintos. Para atributos quantitativos, pode-se discretizar o atributo. Caso isso não seja possível, por restrições de domínio ou perda de precisão, uma possível estratégia é ordenar os exemplos de acordo com os valores do atributo e estabelecer pontos de corte que minimizem a entropia condicional.

Variações do C4.5 também podem ser obtidas com critérios diferente da entropia, como o índice *Gini* (HAN; KAMBER, 2006). Outros critérios de parada incluem interromper o desenvolvimento da árvore quando a variância da classe se tornar inferior a um limiar r_y ou quando a altura for superior a um limiar r_h . Uma versão do C4.5 utilizando entropia e o critério de parada mais simples é apresentada no Algoritmo 3.2.

3.4.3 Modelos de vizinhança

O classificador k vizinhos mais próximos (k -NN, do inglês *k-Nearest Neighbors*) é um classificador baseado em instâncias e similaridade (FIX; HODGES, 1951; FIX; HODGES J. L., 1989). Diferentemente de outros classificadores, o k -NN é singular no sentido de que ele não possui um modelo de classificação explícito. Um “modelo” k -NN pode ser simplesmente uma cópia de todos os exemplos de treinamento. Ao classificar um novo exemplo, o k -NN busca uma descrição do espaço de decisão consultando os exemplos de treinamento armazenados.

O princípio do classificador k -NN é a “regra dos k vizinhos mais próximos”. A hipótese é que, dado um conjunto de exemplos distribuídos sobre o espaço de dados X , a “vizinhança” de um exemplo $x \in X$ estabelecida por uma função de distância apropriada tende a ser ocupada por exemplos que pertencem à mesma classe que x (COVER; HART, 1967), como ilustrado na Figura 15. Desse modo, a classificação de um exemplo x pode ser realizada com base na informação fornecida pelos exemplos conhecidos que são mais similares a x .

Mais especificamente, seja $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$ um conjunto de treinamento. A classificação de um exemplo (x_i, y_i) cuja classe y_i é desconhecida pode ser realizada em três passos:

Algoritmo 3.2: Indutor de árvores de decisão C4.5

```

1 função Árvore = c4.5 (D, Atributos)
  Parâmetros de Entrada:
    D: conjunto de dados rotulados
    Atributos: descrição do espaço de atributos
  Parâmetros de Saída:
    Árvore: uma árvore de decisão
  Referências:
     $H(P)$ : entropia de um particionamento (Definição 3.3)
2 se todos os exemplos em D pertencem à mesma classe então
3   | Árvore  $\leftarrow$  vértice terminal referente à única classe em D
4 senão
5   | H_min  $\leftarrow \infty$ 
6   | para todo Atributo  $\in$  Atributos faça
7     | P  $\leftarrow$  particionamento sobre D de acordo com Atributo
8     | H  $\leftarrow H(P)$ 
9     | se H < H_min então
10    |   | H_min  $\leftarrow H$ 
11    |   | MelhorAtributo  $\leftarrow$  Atributo
12    | fim
13  | fim
14  | Árvore  $\leftarrow$  vértice referente ao atributo MelhorAtributo
15  | para todo valor único x do atributo MelhorAtributo nos exemplos de D faça
16    | Dx  $\leftarrow$  exemplos de D para os quais melhorAtributo = x
17    | Subárvore  $\leftarrow$  c4.5 (Dx)
18    | adicione aresta de Árvore para Subárvore condicionada a
    |   | melhorAtributo = x
19  | fim
20 fim

```

1. com base em uma função de distância $d : X \times X \mapsto \mathbb{R}$, encontre o conjunto de tamanho k que contém os exemplos de D mais semelhantes a x_i . Isto é, encontre o conjunto $N = \{(x_{v1}, y_{v1}), (x_{v2}, y_{v2}), \dots, (x_{vk}, y_{vk})\} \subset D$ tal que $d(x, x_{vi}) \leq d(x, x_{vj})$ para todos $x_{vi} \in N$ e $x_{vj} \in D \setminus N$;
2. com base em uma função de composição $c : Y^k \mapsto Y$, componha as classes $y_{v1}, y_{v2}, \dots, y_{vk}$ e obtenha a classe estimada $\hat{y}_i$. Por exemplo, c pode associar $\hat{y}_i$ ao valor da moda das classes dos exemplos mais próximos de x_i ;
3. assumo $y_i = \hat{y}_i$.

Cover e Hart (1967) mostraram, para o caso teórico $N, k \rightarrow \infty$ e $N \gg k$, que a probabilidade de classificação incorreta do k -NN é no máximo o dobro da probabilidade de classificação incorreta do classificador ótimo de Bayes. É possível verificar empiricamente, entretanto, que mesmo para valores pequenos de k o classificador k -NN produz resultados satisfatórios em

Figura 15 – O princípio dos k vizinhos mais próximos

muitos domínios, particularmente quando os objetos classificados são séries temporais (DING *et al.*, 2008; PRATI; BATISTA, 2012; OATES *et al.*, 2012).

Uma versão simples do k -NN utilizando busca linear no conjunto de treinamento, distância euclidiana e composição pela moda é apresentada no Algoritmo 3.3. Embora seja um modelo de classificação consideravelmente simples, o k -NN tem sido amplamente utilizado pela comunidade de AM, podendo também ser considerado um dos algoritmos mais importantes na mineração de dados (WU *et al.*, 2008). No Capítulo 4 será apresentada uma versão do 1-NN que é flexível com respeito à função de distância (ou similaridade) e, conseqüentemente, independente do tipo de representação das instâncias.

3.5 Indução de classificadores de dados temporais

Nas seções anteriores, os modelos de classificação foram apresentados sob a suposição de que os atributos são independentes e que o indutor é capaz de encontrar hipóteses razoáveis mesmo quando os atributos não são independentes. Entretanto, na classificação de séries temporais, a correlação entre as observações é geralmente um aspecto que não pode ser ignorado (FU, 2011). Geurts (2002) descreve dados temporais como “valores de baixo-nível, cada qual correspondendo ao valor de um atributo em um instante”. O termo “baixo-nível” é empregado com o sentido de que as observações, individualmente, não representam características dos objetos. Por isso, cada qual desses valores, isoladamente, é inútil para um indutor (GEURTS, 2002).

Devido a essa característica peculiar de dados temporais, muitos autores utilizam termos como métodos “clássicos” para se referir a técnicas de mineração de dados nas quais se assume que os exemplos podem ser representados como coleções de atributos que podem ser considerados individualmente pelo indutor. Em contraste, métodos de “mineração de dados temporais” são aqueles que consideram explicitamente a necessidade de se considerar observações como

Algoritmo 3.3: Classificador k vizinhos mais próximos

```

1 função  $\hat{y} = \text{knn}(D, k, x)$ 
  Parâmetros de Entrada:
     $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$ : um conjunto de treinamento
     $k$ : número de vizinhos a serem considerados
     $x$ : o exemplo a ser classificado
  Parâmetros de Saída:
     $y$ : a classe prevista para  $x$ 
  Pré-condições:
     $N \geq k$ 
  Referências:
     $d$ : distância euclidiana (pág. 36, Equação 2.1)
2 Vizinhos  $\leftarrow \emptyset$ 
3 para  $i$  de 1 até  $N$  faça
4   | adicione  $x$  ao conjunto Vizinhos
5   | se número de elementos em Vizinhos  $> k$  então
6   |   |  $r \leftarrow$  o vizinho mais distante de  $x$  em Vizinhos
7   |   | remova  $r$  do conjunto Vizinhos
8   | fim
9 fim
10 Classes  $\leftarrow$  os rótulos dos exemplos no conjunto Vizinhos
11  $\hat{y} \leftarrow$  o valor da moda de Classes

```

conjuntos de valores correlacionados e ordenados.

Métodos de classificação de séries temporais podem ser divididos em duas categorias (DENG *et al.*, 2013):

Baseados em atributos: utilizam os valores de “baixo-nível” para extrair atributos temporais de “maior nível”, os quais podem ser considerados sob uma ótica de mineração de dados “clássica”;

Baseados em instâncias: consideram exemplos conhecidos para classificar um novo exemplo, tipicamente empregando uma função de similaridade.

Dois exemplos de classificadores baseados em instâncias são o k -NN e o SVM—máquina de vetores de suporte. O k -NN é também um classificador “preguiçoso” (tradução literal de *lazy classifier*), pois ele “posterga” a generalização dos exemplos de treinamento até que seja apresentada uma nova instância a ser classificada. O SVM, por outro lado, treina um modelo geométrico que minimiza uma função de perda sobre os exemplos do conjunto de treinamento.

3.5.1 Extração de atributos

Classificadores de séries temporais baseados em atributos estabelecem um espaço de atributos no qual as séries podem ser representadas. Esse processo, denominado **extração de**

atributos ou **caracterização**, envolve a extração de atributos que podem ser divididos em dois grupos: os atributos de primeira ordem são extraídos dos valores das observações, ao passo que os atributos de segunda ordem são baseados em diferenças entre valores próximos (NANOPOULOS; ALCOCK; MANOLOPOULOS, 2001). Um exemplo de atributo de primeira ordem é a obliquidade. A obliquidade é uma medida da assimetria de uma distribuição em torno da média, como mostrado na Figura 16 para uma série temporal $S = (s[1], s[2], \dots, s[n])$. A estatística b_1 ,

$$b_1 = \frac{\frac{1}{n} \sum_{t=1}^n (s[t] - m)^3}{\left[\frac{1}{n-1} \sum_{t=1}^n (s[t] - m)^2 \right]^{3/2}}, \quad (3.11)$$

na qual m é a média amostral da série S (Equação 2.5), é um estimador da obliquidade de S .

Quando aplicados a diferenças de ordem da série, esses atributos se tornam atributos de segunda ordem. A diferença de primeira ordem de S é a série ΔS de tamanho $n - 1$ cujas observações são as diferenças $s[t + 1] - s[t]$; a diferença de segunda ordem é a série $\Delta_2 S$ de tamanho $n - 2$ cujas observações são as diferenças $s[t + 2] - s[t]$ e assim por diante. Generalizando para a k -ésima ordem ($k < n$), definimos $\Delta_k S$ como na Equação 3.12 (KIRCHGÄSSNER; WOLTERS, 2007b).

$$\Delta_k S = (s[1+k] - s[1], s[2+k] - s[2], \dots, s[n] - s[n-k]) \quad (3.12)$$

Figura 16 – Ilustração da obliquidade em torno da média. As duas séries temporais na figura (extraídas do conjunto de dados 50Words) divergem bastante em obliquidade. Ambas as séries estão z -normalizadas, logo possuem média $m = 0$ e desvio padrão $s = 1$. O conceito de obliquidade pode ser observado nos histogramas desenhados no eixo vertical das séries. A distribuição de valores acima da média é aproximadamente igual à distribuição de valores abaixo da média na série mostrada na parte *superior* ($b_1 = -0,2$), enquanto a assimetria da distribuição da série mostrada na parte *inferior* é representada como uma “cauda à esquerda” ($b_1 = 3,1$).

Por exemplo, a média da diferença de k -ésima ordem de S ($k < |S|$) pode ser reescrita como

$$\Delta_k m = \frac{1}{n-k} \sum_{t=1}^{n-k} s[t+k] - s[t]. \quad (3.13)$$

O procedimento geral para se induzir um classificador de séries temporais baseado em atributos consiste em:

1. definir um conjunto de atributos ou características $\{f_1, f_2, \dots, f_M\}$;
2. extrair, para cada série temporal $S_i \in D$, um vetor de atributos;
3. treinar um modelo de classificação com os vetores de atributos.

Para encontrarmos a classe de uma série temporal futura S_x , extraímos o vetor de características de S_x e classificamos essa instância normalmente por meio do modelo induzido no passo 3.

Um exemplo dessa abordagem pode ser encontrado em (NANOPOULOS; ALCOCK; MANOLOPOULOS, 2001). Os autores empregaram os quatro primeiro momentos em torno do centro da série—média, desvio padrão, obliquidade e curtose—para extrair características de primeira ordem. Os mesmos momentos, extraídos da diferença de D -ésima ordem das séries, no qual D é um parâmetro definido pelo usuário, foram utilizados como características de segunda ordem. De cada série temporal foram extraídas, portanto, oito características. Um modelo de redes neurais foi treinado com diferentes configurações e avaliado em conjuntos de séries temporais artificialmente produzidas para simular cartas de controle—dados temporais de “parâmetros críticos” empregados para detectar anormalidades no funcionamento de sistemas complexos. Os autores compararam essa abordagem contra modelos de redes neurais treinados com as instâncias das séries, argumentando que resultados favoráveis são obtidos pelo classificador induzido dos atributos.

Não é necessário que os atributos sejam extraídos diretamente das observações ou de diferenças das observações. Atributos de qualquer ordem podem ser extraídos, por exemplo, de representações alternativas das séries e de subsequências produzidas por uma janela deslizante ou um processo de segmentação. Considere os seguintes casos de extração de atributos temporais:

Extraídos de domínios alternativos: a série é transformada para um domínio alternativo de representação, no qual os atributos podem ser efetivamente extraídos. Por exemplo, utilizando a transformada discreta de Fourier (Seção 5.2) é possível identificar, em unidades de frequências conhecidas, as componentes periódicas contidas em um sinal digital. Essas componentes podem ser utilizadas como atributos de frequências do sinal.

Um tipo de aplicação no qual atributos de frequência podem ser utilizados é o reconhecimento de fala (GANCHEV; FAKOTAKIS; KOKKINAKIS, 2005). Os coeficientes de

frequência Melcepstrais (MFCC, *Mel-frequency cepstral coefficients*) são obtidos de uma transformação dos coeficientes de frequência representados na escala Mel. A escala Mel é uma escala de frequências que se aproxima da percepção auditória humana e pode ser utilizada para extrair características de sinais digitais que são relevantes na detecção de padrões associados à fala. Sistemas de telefonia empregam coeficientes MFCC para reconhecer dígitos falados, substituindo a necessidade de se utilizar o teclado numérico do telefone para “navegar” entre menus de um sistema de atendimento (BRESOLIN; NETO; ALSINA, 2008).

Extraídos de segmentos: o processo de extração envolve segmentar a série temporal. Pode-se extrair um conjunto de atributos de cada segmento ou combinar coeficientes de cada segmento em atributos de maior ordem. A representação PAA (Seção 5.3) é obtida segmentando-se uma série temporal em N subsequências e extraíndo-se os valores médios das observações de cada uma delas.

Extraídos de uma janela deslizante: o processo envolve uma janela de tamanho $w + 1$ que seleciona as subsequências $s[1 : 1+w]$, $s[2 : 2+w]$, ..., $s[n-w : n]$. Para cada posição da janela, pode-se extrair uma coleção de coeficientes. Alguns autores estabelecem uma área de exclusão a fim de reduzir ou eliminar casamentos triviais entre as subsequências sob a janela, deslizando a janela em passos $w_s > 1$.

Muitas vezes uma técnica de extração de atributos envolve mais de um desses casos. Por exemplo, a representação baseada em *wavelets* é obtida quando uma janela deslizante é associada a uma transformada de frequências. Essa representação será abordada na Seção 5.2.4 desta Tese.

3.5.2 Classificação baseada em instâncias

Classificadores baseados em instância utilizam-se de uma função de similaridade que elimina a necessidade de caracterizar os objetos em atributos. No contexto de dados temporais, isso pode ser uma propriedade bastante desejável, pois encontrar atributos representativos de dados temporais é uma tarefa não trivial, devido principalmente à forte dependência entre as observações e à alta dimensionalidade dos dados.

Um dos exemplos mais proeminentes de classificadores baseados em instâncias é o 1-NN (Seção 3.4.3). Em que pese sua simplicidade, esse modelo vem sendo repetidamente confirmado como “notório” na classificação de dados temporais, particularmente quando associado à distância DTW (FARD *et al.*, 2016; LINES; BAGNALL, 2015; BAGNALL; LINES, 2014). Uma possível explicação para isso se deve ao fato de que a distância euclidiana considera todas as observações das séries e a relação de ordem entre as observações de duas séries em um único fator. E a distância DTW pode ser considerada uma “variação” da distância euclidiana no qual esta é associada a invariância a *warping* local.

Ao longo desta Tese, trataremos de diversos casos de classificação baseada em instâncias. No Capítulo 5, serão considerados classificadores de séries temporais baseados em instâncias transformadas para domínios alternativos de representação. E no Capítulo 4 serão considerados classificadores de séries temporais utilizando diferentes critérios de similaridade.

3.5.3 Árvores de Shapelets

O conceito de *shapelet* foi proposto por Ye e Keogh (2009) como uma nova primitiva de séries temporais, juntamente com um algoritmo para construção de árvores de decisão baseadas em *shapelets*. Uma árvore de *shapelets* é uma espécie de árvore de decisão na qual cada vértice não terminal está associado a um *shapelet*. As arestas são condições de distância entre o *shapelet* e as séries. Nesse sentido, o *shapelet* tem papel semelhante ao de um atributo “implícito”. Para se construir uma árvore de decisão com *shapelets*, devemos primeiramente encontrar esse atributo “implícito”, de modo que os consequentes valores assinalados ao exemplo por esse atributo permitam realizar uma separação adequada do espaço de decisão.

Em sua proposição original, um *shapelet* é necessariamente um atributo que minimiza a entropia de um particionamento imposto aos exemplos de treinamento. Nesta Tese, adota-se uma definição mais genérica e compatível com trabalhos nos quais *shapelets* são obtidos de modo aproximado ou nos quais obtém-se mais de um *shapelet* por classe para definir espaços de atributos (GORDON; HENDLER; ROKACH, 2012; YANG *et al.*, 2016; CAVALCANTE, 2016).

Definição 3.4: Seja $D \subset \mathbb{T}$ um domínio de séries temporais composto por séries potencialmente de diferentes tamanhos. Um *shapelet* é uma subsequência $\check{S} = \{\check{s}[1], \dots, \check{s}[n]\}$ associada a um limiar $r \in \mathbb{R}$ que produz um particionamento $P_D = \{T_1, T_2\}$ tal que

$$d(S_j, \check{S}) < r \leq d(Z_k, \check{S})$$

para todas as séries $S_j \in T_1$ e $Z_k \in T_2$.

Note que, para um *shapelet* $\check{S}$ e um conjunto de N séries temporais, existem $N-1$ particionamentos possíveis que respeitam a relação Definição 3.4. A fim de encontrar o melhor particionamento, definimos o conceito de ponto de corte ótimo:

Definição 3.5: Seja $\check{S}$ um *shapelet* e $D \subset \mathbb{T}$ um conjunto de treinamento. Sejam $P_1, P_2, \dots, P_{N-1}$ todos os possíveis particionamentos que podem ser impostos em D por $\check{S}$. Um ponto de corte ótimo para o *shapelet* $\check{S}$ é o limiar $r \in \mathbb{R}$ associado ao particionamento de menor entropia.

Uma forma simples de se encontrar o ponto de corte ótimo para um *shapelet* é calcular todas as distâncias $d_1 = (S_1, \check{S})$, $d_2 = (S_2, \check{S})$, $\dots$, $d_N = (S_N, \check{S})$ para as séries $S_i \in D$. A distância entre um *shapelet* $\check{S}$ e uma série temporal é a distância entre a subsequência $\check{S}$ e a série. Assumindo-se, sem perda de generalidade, que as distâncias foram calculadas de modo que a

sequência de valores $d_1, d_2, \dots, d_N$ é crescente, então pode-se testar todos os pontos de corte entre as margens de separação, $p_i = \frac{d_i + d_{i+1}}{2}$, até encontrar o ponto de corte ótimo.

Para um problema de classificação binária, uma árvore de *shapelets* pode ser representada por um único *shapelet* $\check{S}_1$ e seu ponto de corte ótimo r_1 . O nó terminal esquerdo é associado à classe positiva e o nó direito à negativa. Para classificar uma nova instância x , basta verificar se a distância entre a série e o *shapelet* é inferior ao limiar r_1 . Nesse caso a instância deve ser classificada como positiva; caso contrário ela pertence à classe negativa.

Para problemas de classificação multiclases, pode-se gerar uma nova árvore para os exemplos de cada sub-espaco induzido pelo *shapelet* do nível superior de maneira recursiva, assim como faz o indutor C4.5. A indução pode prosseguir até que o espaço produzido por um *shapelet* contenha apenas séries de uma mesma classe ou outro critério de parada seja alcançado (GORDON; HENDLER; ROKACH, 2012). Esse procedimento geral é mostrado no Algoritmo 3.4.

Algoritmo 3.4: Indução de árvores de *shapelets*.

1 **função** $\text{Árvore} = \text{ShapeletTree}(D)$ **Parâmetros de Entrada:**

$D = \{(S_1, y_1), (S_2, y_2), \dots, (S_n, y_n)\}$: conjunto de séries rotuladas

Parâmetros de Saída:

Árvore: uma árvore de *shapelets*

Referências:

MelhorShapelet: algoritmo que retorna o melhor *shapelet* e seu respectivo limiar para um $\hookrightarrow$ conjunto de séries temporais

2 **se** todos os exemplos em D pertencem à mesma classe **então**

3 | Árvore $\leftarrow$ vértice terminal referente à única classe em D

4 **senão**

5 | $(\check{S}, r) \leftarrow \text{MelhorShapelet}(D)$

6 | $D_e \leftarrow \{S_j \in D \text{ tais que } d(S_j, \check{S}) < r\}$

7 | $D_d \leftarrow \{S_k \in D \text{ tais que } d(S_k, \check{S}) \geq r\}$

8 | Árvore $\leftarrow$ vértice associado ao par $(\check{S}, r)$

9 | RamoEsquerdo $\leftarrow \text{ShapeletTree}(D_e)$

10 | RamoDireito $\leftarrow \text{ShapeletTree}(D_d)$

11 | Adicione aresta de Árvore para RamoEsquerdo condicionada a $d(\cdot, \check{S}) < r$

12 | Adicione aresta de Árvore para RamoDireito condicionada a $d(\cdot, \check{S}) \geq r$

13 **fim**

Resta estabelecer o algoritmo MelhorShapelet, a função que identifica o melhor *shapelet* e o limiar associado para um conjunto de séries temporais. Na definição adotada nesta Tese, qualquer subsequência pode ser um *shapelet*, contanto que tenha um número de observações compatível com o domínio—isto é, o *shapelet* não pode ser mais longo que a menor série temporal do domínio. Uma abordagem trivial é enumerar todas as possíveis subsequências do conjunto de treinamento e escolher o melhor *shapelet*. Esse algoritmo de força bruta pode ser

sumarizado nos seguintes passos:

- De um conjunto de N séries temporais com tamanho médio n , selecionar uma série S_i ;
- Enumerar todas as subsequências de tamanho médio m ;
- Para cada uma das $O(n^2)$ subsequências de S_i , calcular sua distância para todas as séries do conjunto e encontrar um ponto de corte ótimo;
- Se o *shapelet* extraído de S_i for melhor do que o melhor *shapelet* encontrado até o momento, descartar o *shapelet* anterior e armazenar o novo *shapelet*;
- Procurar *shapelets* nas séries restantes do conjunto.

O cálculo da distância entre uma subsequência e uma série temporal pode ser conduzido em $O(mn)$, no qual m é o tamanho da subsequência e n é o tamanho da série. Note que $O(mn) = O(n^2)$. O algoritmo para encontrar o ponto de corte ótimo, apresentado no início desta seção, calcula a distância entre o *shapelet* e todas as séries do conjunto de treinamento e possui complexidade $O(Nn^2)$ por subsequência. Portanto, o custo para estimar a utilidade de todos os $O(n^2)$ *shapelets* que podem ser extraídos de uma série é $O(Nn^4)$, de modo que a complexidade de tempo total do algoritmo de força bruta é $O(N^2n^4)$. Claramente, é uma abordagem inviável.

Para reduzir o custo computacional, Ye e Keogh (2009) propuseram diversas podas no espaço de busca utilizando abandono antecipado do cálculo da distância e da entropia. Embora a complexidade assintótica desse algoritmo³ seja também $O(N^2n^4)$, no caso médio ele pode ser mais eficiente que o algoritmo de força bruta em várias ordens de magnitude (YE; KEOGH, 2009).

Utilizando uma técnica de memoização (MUEEN; NATH; LIU, 2010), é possível reduzir a complexidade de tempo da avaliação da qualidade de cada *shapelet* para $O(Nn)$. Esse ganho em tempo é obtido ao custo de uma complexidade espacial maior— $O(Nn^2)$ —, mas permite reduzir a complexidade total do algoritmo de *shapelets* para $O(N^2n^3)$. Mueen, Keogh e Young (2011) associam essa estratégia a uma poda do espaço de busca que utiliza a desigualdade triangular para reduzir o número de passos necessários para estimar se um *shapelet* particiona o espaço de decisão com menor entropia que os *shapelets* encontrados até o momento.

3.6 Erro e acurácia de um classificador

Uma importante questão no aprendizado indutivo é: dado que um classificador é um modelo induzido de uma amostra dos dados, como se pode avaliar o grau de fidelidade do classificador ao conceito desejado? Uma forma de se mensurar essa fidelidade é testar o classificador

³ Os autores afirmam que a distância entre uma subsequência e uma série pode ser realizado em $O(n)$, mas não consideram o custo necessário para z -normalizar cada subsequência da série na busca do casamento ideal entre esta e a subsequência.

e verificar com que frequência ele produz resultados incompatíveis com a função-conceito. Em outras palavras, para quais exemplos x_i observamos o resultado $\hat{h}(x_i) \neq \zeta(x_i)$?

Por definição, é razoável assumir que tanto o espaço de exemplos quanto a função-conceito são desconhecidos. Isso quer dizer que não apenas o mapeamento $\zeta(x_i)$ é difícil de ser caracterizado, mas o mapeamento $\hat{h}(x_i)$ também não pode ser enumerado. Além disso, a informação $\hat{h}(x_i)$ é insuficiente, pois não leva em consideração quais exemplos x_i serão observados.

Com base nessas observações, o questionamento anterior pode ser reformulado para o seguinte: “qual a probabilidade de que um modelo $\hat{h}$ seja incompatível com a função-conceito ζ , dado que um exemplo x_i foi observado?” Considerando-se que os exemplos são extraídos de uma distribuição $\mathcal{D}$, define-se o **erro verdadeiro** de um classificador (MITCHELL, 1997).

Definição 3.6: O erro verdadeiro de um classificador $\hat{h}$ com respeito à uma função-conceito ζ e uma distribuição de exemplos $\mathcal{D}$ é a probabilidade de que o resultado de classificação $\hat{h}(x)$ seja incompatível com $\zeta(x)$:

$$\text{err}_{\mathcal{D}}(\hat{h}) \equiv P_{x \in \mathcal{D}} (\hat{h}(x) \neq \zeta(x)). \quad (3.14)$$

É em geral impossível conhecer o erro verdadeiro, porém é possível *estimá-lo*. Dada uma amostra S de exemplos obtidos da distribuição $\mathcal{D}$, o **erro amostral** do classificador $\hat{h}$ é dado por

$$\text{err}_S(\hat{h}) = \frac{1}{|S|} \sum_{x \in S} \mathbb{I}[\hat{h}(x) \neq \zeta(x)], \quad (3.15)$$

no qual $\mathbb{I}[\cdot] \in \{0, 1\}$ é a função indicadora e $\mathbb{I}[aRb] = 1$ se e somente se a relação aRb é válida.

Uma forma simples de se calcular o erro amostral é testar o modelo com os próprios exemplos de treinamento. Esse procedimento é denominado **ressubstituição** (MONARD; BARANAUSKAS, 2003) e o estimador obtido é o **erro empírico** ou **erro aparente** (ALPAYDIN, 2010). Seja $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$ um conjunto de treinamento, então o erro aparente de um modelo $\hat{h}$ é a fração

$$\text{err}_D(\hat{h}) = \frac{1}{N} \sum_{i=1}^N \mathbb{I}[\hat{h}(x_i) \neq y_i]. \quad (3.16)$$

Entretanto, esse estimador é afetado por um viés entre o modelo $\hat{h}$ e a relação $\hat{h}(x_i) \neq y_i$. Afinal, o indutor seleciona $\hat{h}$ visando maximizar uma função de utilidade que de alguma forma envolve classificar corretamente os exemplos de treinamento. Dessa forma, o erro aparente é uma visão otimista do erro verdadeiro.

Uma “regra fundamental” do aprendizado indutivo é que qualquer avaliação de um modelo deve ser conduzida com amostras separadas para treinamento e teste. Uma estratégia simples, denominada **holdout**, consiste em separar uma porção $p \cdot N$ (com $0 < p < 1$) dos

Figura 17 – Exemplo de reamostragem e *holdout*.

exemplos rotulados para induzir um modelo $\hat{h}_r$ e utilizar os $(1 - p) \cdot N$ exemplos restantes para os testes, como ilustrado na Figura 17.

Através do *holdout*, o erro amostral do classificador pode ser estimado, dado um conjunto de teste $T = \{(x_{t_1}, y_{t_1}), (x_{t_2}, y_{t_2}), \dots, (x_{t_M}, y_{t_M})\}$, como sendo a fração

$$\text{err}(\hat{h}_r) = \frac{1}{M} \sum_{i=1}^M \mathbb{I}[\hat{h}_r(x_{t_i}) \neq y_{t_i}], \quad (3.17)$$

cujo valor se aproxima do erro verdadeiro para um conjunto de testes representativo e suficientemente grande (WEISS; KULIKOWSKI, 1991).

Outra importante medida de qualidade de um classificador é a **acurácia**. A acurácia é o “inverso” do erro e diz respeito aos exemplos corretamente classificados pela hipótese. Para qualquer modelo $\hat{h}$ e conjunto de exemplos $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$, a acurácia é a fração

$$\text{acc}(\hat{h}) = 1 - \text{err}(\hat{h}) = \frac{1}{N} \sum_{i=1}^N \mathbb{I}[\hat{h}(x_i) = y_i]. \quad (3.18)$$

A acurácia e o erro são visões distintas de um mesmo conceito de qualidade: ambas dizem respeito ao acerto ou erro do classificador “no caso médio”. Outras medidas de qualidade incluem a precisão, o *recall* e a F-1. Estas são mais tipicamente utilizadas em problemas de classificação binária, nos quais a classe positiva é considerada “mais relevante”. Por exemplo, suponha que à classe positiva pertencem os *emails* que contêm *spam*. A **precisão** (Equação 3.18) mede o grau de confiança que pode ser depositado no classificador para os *emails* declarados como *spam* e o **recall** (Equação 3.20) mede a taxa de *spam* que o classificador é capaz de capturar.

$$\text{prec}(\hat{h}) = \frac{\sum_{i=1}^N \mathbb{I}[\hat{h}(x_i) = \oplus] \cdot \mathbb{I}[y_i = \oplus]}{\sum_{i=1}^N \mathbb{I}[\hat{h}(x_i) = \oplus]} \quad (3.19)$$

$$\text{rec}(\hat{h}) = \frac{\sum_{i=1}^N \mathbb{I}[\hat{h}(x_i) = \oplus] \cdot \mathbb{I}[y_i = \oplus]}{\sum_{i=1}^N \mathbb{I}[y_i = \oplus]} \quad (3.20)$$

O classificador ideal possui precisão e *recall* máximos. Entretanto, esses objetivos são em geral conflitantes. Um indutor pode maximizar a precisão para os exemplos de treinamento, mas produzir um modelo que generaliza pouco e não consegue classificar exemplos positivos muito diferentes daqueles utilizados na indução—baixa *recall*. Ou o indutor pode produzir um modelo menos específico aos exemplos de treinamento, o que acarreta maior capacidade de generalização, mas tende a um maior erro na classe positiva—baixa precisão. A medida F-1 (Equação 3.21) é a média harmônica da precisão e do *recall*, podendo ser empregada como uma tentativa de se obter um equilíbrio entre ambas. O caso particular em que precisão e *recall* são considerados igualmente é chamado F-1 e é dado pela Equação 3.21.

$$\text{F1}(\hat{h}) = 2 \frac{\text{prec}(\hat{h}) \cdot \text{rec}(\hat{h})}{\text{prec}(\hat{h}) + \text{rec}(\hat{h})} \quad (3.21)$$

Essas e outras medidas de qualidade podem ser generalizadas para problemas de classificação multiclasse, considerando-se as taxas de verdadeiro-positivo, falso-positivo, verdadeiro-negativo e falso-negativo (SOKOLOVA; LAPALME, 2009). Para cada classe C_j , a taxa de falso-positivo é a fração dos exemplos que pertencem à classe C_j (positivo), mas que o classificador identificou como pertencentes a outra classe qualquer (falso). Os demais conceitos generalizam-se dessa relação⁴. Nesse caso, pode-se avaliar um classificador com respeito à sua precisão (ou *recall*) em uma classe específica, à soma total da qualidade do classificador em todas as classes ou à média para todas as classes.

Como será mostrado ao longo desta Tese, os trabalhos mais próximos deste projeto utilizam principalmente as medidas de erro ou acurácia na avaliação dos classificadores de séries temporais. Uma vez que o objetivo deste trabalho é, principalmente, explorar como as diferentes representações de séries temporais podem ser utilizadas para aproximar adequadamente a função-conceito no “caso médio”, os classificadores serão avaliados com respeito a essas medidas. Mais especificamente, será utilizada a acurácia.

⁴ Extrapolando: falso-positivo é a taxa de exemplos incorretamente classificados como $\hat{h}(x) = C_j$; verdadeiro-negativo: corretamente classificados como $\hat{h}(x) \neq C_j$; e falso-negativo: incorretamente classificados como $\hat{h}(x) \neq C_j$.

3.6.1 Validação por reamostragem

O *holdout* efetivamente elimina o viés inerente ao erro empírico, mas cria um dilema secundário: qual fração p deve ser reservada para testar o modelo? Idealmente, desejamos utilizar o maior número possível de exemplos no treinamento, a fim de produzir um modelo $\hat{h}_r$ bastante semelhante a $\hat{h}$. Por outro lado, se o conjunto de testes for pequeno, a variância do estimador será alta (MITCHELL, 1997).

Na ausência de abundância dos dados, pode-se recorrer a estratégias de validação por reamostragem. O termo **reamostragem** diz respeito a um conjunto de práticas que visam reutilizar uma amostra limitada, produzindo vários estimadores que na média se aproximam da acurácia ou erro que seria obtido caso um vasto conjunto de exemplos de teste estivesse disponível. Em linhas gerais, procedimentos de reamostragem são conduzidos como ilustrado na Figura 18. De uma amostra S , são sub-amostrados vários conjuntos de treinamento e validação, dos quais são induzidos os modelos $\hat{h}_1, \hat{h}_2, \dots, \hat{h}_k$. Os modelos são testados isoladamente nos conjuntos de validação, produzindo k estimadores $\text{acc}(\hat{h}_1), \dots, \text{acc}(\hat{h}_k)$.

Figura 18 – Ilustração do processo de reamostragem na validação de classificadores.

Em todos os casos, o conjunto de treinamento D_i é disjuncto do conjunto de validação T_i . A acurácia (ou outra medida) esperada para o classificador $\hat{h}$ é estimada como a média da acurácia obtida pelos classificadores $\hat{h}_1, \dots, \hat{h}_k$:

$$\text{acc}_S(\hat{h}) = \frac{1}{k} \sum_{i=1}^k \text{acc}(\hat{h}_i). \quad (3.22)$$

O *holdout* pode ser considerado um caso particular de reamostragem no qual $D_1 \cup T_1 = S$. Como explicado no início desta seção, o emprego desse método envolve um compromisso delicado entre o viés e a variância do estimador.

Uma estratégia derivada do *holdout* é a *subamostragem aleatória* ou, simplesmente, **amostragem aleatória**. A amostragem aleatória nada mais é do que k repetições do *holdout* (HAN; KAMBER, 2006). Embora produza estimadores menos enviesados que o *holdout*, a amostragem aleatória introduz dependência na distribuição das classes dos conjuntos de treinamento e validação (KOHAVI, 1995). Como cada par (D_i, T_i) é um particionamento de D , uma classe super-representada em D_i será necessariamente sub-representada em T_i .

Os métodos de *bootstrap* podem ser utilizados como alternativas à amostragem aleatória. A premissa fundamental do *bootstrap* é a suposição de que a amostra S é o conjunto mais representativo possível da distribuição $\mathcal{D}$. O *bootstrap* reamostra S com reposição a fim de produzir distribuições aproximadas de S .

Existem diversos métodos de reamostragem *bootstrap*, sendo o mais comum em AM o *bootstrap* e_0 (WEISS; KULIKOWSKI, 1991). Em cada passo, um processo de reamostragem *bootstrap* e_0 seleciona $N = |S|$ exemplos de S para compor o conjunto de treinamento. Isso significa que D_i é um multiconjunto no qual uma instância x_i pode ocorrer mais de uma vez. O complemento desse multiconjunto compõe o conjunto de validação $T_i = S \setminus D$. Uma propriedade do *bootstrap* e_0 é que a probabilidade de um exemplo x_i não ser selecionado após N passos é $(1 - 1/N)^N \approx e^{-1} \approx 0,368$. Em cada iteração, aproximadamente 63,2% dos exemplos são utilizados na validação do modelo.

3.6.2 Validação cruzada

Validação cruzada é um conjunto de métodos de reamostragem baseados em particionamentos sistemáticos do conjunto S . Métodos de validação cruzada asseguram que cada exemplo será utilizado pelo menos uma vez na validação do modelo ao mesmo tempo em que buscam maximizar a cardinalidade dos conjuntos de treinamento, minimizando o viés do estimador.

O *k-fold cross-validation* (*k-fold CV*) é um procedimento de reamostragem no qual as amostras são particionadas aleatoriamente em k subconjuntos disjuntos $F_1, F_2, \dots, F_k$ denominados *folds*. Todos os *folds* contêm aproximadamente o mesmo número de exemplos e sua união é o conjunto D (HAN; KAMBER, 2006). Em cada iteração, um *fold* F_i é utilizado como o conjunto de validação e os demais são combinados para compor o conjunto de treinamento, sobre o qual se induz um modelo $\hat{h}_i$. É comum que o particionamento do *k-fold CV* seja **estratificado**, de modo que a proporção das classes em cada *fold* é aproximadamente igual à distribuição das classes em D .

No *k-fold CV*, cada exemplo é utilizado uma única vez na validação do modelo e $k-1$ vezes no treino dos classificadores $\hat{h}_1, \dots, \hat{h}_k$. O viés do estimador produzido pelo *k-fold CV* diminui conforme o número de *folds* cresce, ao passo em que a variância aumenta devido à correlação entre os conjuntos de treinamento, pois cada par de *folds* compartilha $N \cdot \frac{k-2}{k}$ exemplos. Um compromisso razoável é tipicamente $k = 10$, caso particular em que o procedimento é

denominado *10-fold cross-validation* (*10-fold CV*).

Quando o número de *folds* é igual ao número de exemplos, obtém-se validação cruzada *leave-one-out*. Em cada iteração, um exemplo x_i é retirado da amostra e o classificador $\hat{h}_i$ é induzido com os exemplos restantes. O estimador obtido pelo *leave-one-out* normalmente possui viés menor que outros procedimentos de validação cruzada. Porém, como a variância desse estimador é elevada, essa estratégia de validação não é recomendada para amostras pequenas. Similarmente, observe que a variância do *10-fold CV* é inversamente proporcional ao tamanho da amostra e essa estratégia também é inadequada para casos em que há poucos exemplos disponíveis. No caso extremo em que $N = 10$, o *10-fold CV* é reduzido ao *leave-one-out*.

Uma alternativa para amostras pequenas é a realização múltipla do processo de validação cruzada. O 5×2 *cross-validation* (5×2 CV) é o caso no qual o estimador $\text{acc}(\hat{h})$ é obtido da média de cinco realizações de *2-fold cross-validation* (DIETTERICH, 1998). O 5×2 CV é um procedimento de validação cruzada no qual os números de exemplos nos conjuntos de treinamento e de validação são simultaneamente máximos. O número de replicações foi obtido empiricamente por Dietterich (1998). De acordo com o autor, quando um número inferior de repetições é empregado, o viés de estimativa do *2-fold CV* afeta o resultado negativamente; quando um número superior de repetições é empregado, é a dependência entre as replicações do *2-fold CV* que introduz viés considerável. É possível que, para uma amostra S específica, uma estratégia semelhante envolvendo outro número de replicações ou *folds* seja mais adequada, mas esse problema não é abordado nesta Tese.

3.7 Comparação de indutores

Uma importante questão do aprendizado de máquina é comparar diferentes algoritmos de aprendizado segundo algum critério de qualidade. Considerando-se como critério a fidelidade do modelo com respeito à função-conceito, duas perguntas comuns são:

1. Existem aplicações específicas nas quais utilizar um indutor, em detrimento dos demais, pode produzir modelos de classificação com maior acurácia?
2. Existe algum indutor que seja capaz de produzir classificadores com maior acurácia, consistentemente, em diversos domínios de aplicação?

A resposta para a primeira pergunta é pontual. Nesta Tese, a utilidade de algum domínio de representação ou função de distância de dados temporais é justificada pela existência de um número “substancial” de conjuntos de dados nos quais o uso de representações alternativas permite obter classificadores mais confiáveis do que o 1-NN no domínio temporal. Por exemplo, como será mostrado no Capítulo 4, existem casos particulares nos quais algumas funções de distância “pouco convencionais” produzem classificadores mais apropriados do que as funções

de distância tipicamente encontradas em trabalhos do “estado-da-arte”. A significância disso é pautada em dois argumentos: muitas vezes tendemos a focar nas estratégias adotadas pelo estado-da-arte quando, para aplicações específicas, técnicas normalmente negligenciadas poderiam fornecer resultado superiores. Ademais, a existência de estratégias que são sub-ótimas no caso médio, mas superiores em casos específicos, sugere sua utilidade na composição de *ensembles* de classificadores.

A resposta para a segunda pergunta normalmente envolve empregar dois ou mais indutores em diversos conjuntos de dados e comparar os valores de acurácia dos modelos produzidos. O repositório UCR se destaca nesse sentido, pois fornece um grande número de conjuntos de dados temporais rotulados que abrangem vários domínios de conhecimento e variam significativamente em termos de “dificuldade” do problema de classificação (número de classes, desbalanceamento do conjunto de treinamento etc.).

Em ambos os casos, é importante adotar uma estratégia de avaliação adequada. Como mencionado na Seção 2.6.1, os conjuntos de dados de *benchmark* do repositório UCR são bastante utilizados em trabalhos comparativos de classificação de dados temporais. Esses conjuntos são distribuídos com um particionamento prévio de exemplos de treinamento e teste. Esse particionamento, que será doravante denominado o *particionamento canônico do repositório UCR*, nada mais é do que uma única realização do método de reamostragem *holdout* para cada conjunto de dados.

Uma decisão de pesquisa tomada no início deste trabalho foi seguir o particionamento canônico do repositório UCR nos experimentos. Como mencionado anteriormente neste capítulo, essa decisão acarreta na inserção de um viés, devido ao fato que o *holdout* é um caso único de reamostragem. Por outro lado, seguir o particionamento canônico do repositório UCR tem como vantagem tornar os resultados obtidos diretamente comparáveis com outros trabalhos (KATE, 2016), bem como a redução do custo computacional associado aos experimentos. Além disso, foi levada em consideração a declaração de um dos pesquisadores envolvidos na manutenção do repositório UCR de que misturar exemplos de treinamento e teste do particionamento canônico introduz viés em alguns conjuntos de dados (BAGNALL *et al.*, 2012). Ademais, é esperado que o viés associado ao *holdout* tenha efeito mínimo em experimentos nos quais se promove comparação de um grande número de algoritmos sobre elevado número de conjunto de dados. É improvável que um particionamento aleatório seja sistematicamente favorável a algum algoritmo específico em detrimento de muitos outros.

Desse modo, para a apresentação dos resultados finais de cada indutor, optou-se por manter o particionamento canônico do repositório UCR. Não obstante, em muitos casos é importante comparar a acurácia do classificador antes e depois da utilização dos exemplos de teste. Por exemplo, quando o classificador possui parâmetros, pode ser necessário encontrar valores ideais para cada conjunto de dados. Neste trabalho, optou-se por utilizar reamostragem *leave-one-out* para tais situações.

Para fins de clareza, serão adotados neste texto dois termos para indicar os valores de acurácia obtidos por um classificador quando avaliado sobre os conjuntos de treinamento e sobre os conjuntos de teste. O termo **acurácia esperada** será utilizado para designar o valor da acurácia de um classificador avaliado com reamostragem dos exemplos de treinamento⁵. O termo **acurácia obtida** será utilizado para designar o valor de acurácia observada nos exemplos de teste do particionamento canônico do repositório UCR. A diferença entre a acurácia esperada e a acurácia obtida é um importante estimador da capacidade de se prever o desempenho de um classificador antes de empregá-lo em uma situação de “mundo real”.

Com base nesses conceitos, podemos estabelecer diferentes métodos que nos permitem inspecionar e comparar diferentes algoritmos de indução.

3.7.1 Comparação por meio de gráficos

Ferramentas visuais são extremamente úteis na comparação de algoritmos. O gráfico de espalhamento é uma ferramenta visual que pode ser utilizada para comparar o desempenho de dois algoritmos sobre vários casos de teste. Seu formato é simples e ele permite fácil interpretação. Um exemplo é dado na Figura 19, na qual se compara o classificador 1-NN utilizando a distância DTW (Seção 4.6.1) contra o mesmo algoritmo baseado em distância euclidiana. As coordenadas de cada ponto representam simultaneamente a acurácia obtida por meio da distância DTW e da distância euclidiana.

Figura 19 – Exemplo de gráfico de espalhamento.

O conjunto Phoneme, como mencionado anteriormente, é o conjunto mais “difícil” do

⁵ Note que a acurácia esperada é distinta da acurácia aparente, pois é estimada por um processo no qual cada estimador individual deriva de um classificador induzido e testado em amostras separadas dos exemplos de treinamento.

repositório UCR e não é surpreendente que o ponto associado a esse conjunto esteja bastante próximo da origem—tanto a distância euclidiana quanto a DTW falham em classificar corretamente mais de 25% dos exemplos. Já o conjunto MiddlePhalanxOutlineCorrect é um caso no qual a distância euclidiana produz resultados melhores que a distância DTW. Além disso, é o caso no qual a distância euclidiana mais se destaca com relação à DTW, como pode ser facilmente observado na figura devido ao seu posicionamento na região não sombreada.

O gráfico de espalhamento é uma ferramenta visual bastante útil que fornece uma visão geral do desempenho médio de dois classificadores. Quanto mais os pontos se afastam da diagonal, mais forte é a evidência de que um dos algoritmos é superior ao outro. Em alguns casos, como na Figura 19, a predominância de pontos em uma ou outra região é tão grande que “dispensa” testes adicionais para a maioria das argumentações. Apesar disso, o gráfico de espalhamento padece de uma falha: ele não permite prever qual algoritmo será mais eficiente antes da avaliação. Isso pode levar a uma falácia informal denominada “falácia do atirador texano”. O nome dessa falácia (*Texan sharpshoot fallacy*) advém de uma anedota na qual um atirador efetua disparos de rifle contra a parede de um celeiro e então desenha um alvo ao redor dos buracos, causando a impressão de ser um atirador extremamente preciso.

Embora o atirador da anedota seja claramente mal-intencionado, a analogia dessa falácia com a pesquisa científica não exige nem presume malícia. Comete-se essa falácia quando se atribui significância estatística a resultados aleatórios observados em um contexto limitado (THOMPSON, 2009). Em outras palavras, simplesmente contar o número de resultados superiores obtidos por um algoritmo pode ser insuficiente e induzir ao erro.

Batista, Wang e Keogh (2011) defendem que é improdutivo mostrar que um algoritmo obtém resultados superiores a menos que seja possível *prever* que os resultados serão superiores. Se pudermos prever qual algoritmo será superior, então podemos sempre optar pelo melhor algoritmo. Caso contrário, correremos o risco de utilizar o algoritmo inferior para tentar solucionar um problema. Para comparar dois algoritmos, os autores propuseram um gráfico que eventualmente ficou conhecido como *sharpshoot plot*—**gráfico sharpshoot**⁶. O gráfico *sharpshoot* é um gráfico de espalhamento no qual um eixo representa o **ganho esperado** e o outro representa o **ganho obtido** por um algoritmo de “interesse” sobre um algoritmo de “referência”. O conceito de ganho é definido simplesmente como a razão

$$\text{ganho} = \frac{\text{acurácia do algoritmo de interesse}}{\text{acurácia do algoritmo de referência}},$$

de modo que os ganhos esperado e obtido são calculados com base nos conceitos de acurácia esperada e obtida apresentados na seção anterior.

⁶ O autor desta Tese desconhece a existência de um termo em português para este gráfico que tenha aceitação na comunidade científica. Portanto, o termo “gráfico *sharpshoot*” será utilizado como um neologismo.

Um exemplo de gráfico *sharpshoot* é dado na Figura 20. Cada ponto representa um dos conjuntos de dados do repositório UCR. O algoritmo de interesse é o 1-NN com distância DTW e o algoritmo de referência é o classificador 1-NN com distância euclidiana. Trata-se dos mesmos resultados mostrados na Figura 19, porém com a informação adicional de acurácia esperada.

Figura 20 – Exemplo de gráfico *sharpshoot*. O algoritmo de interesse é o 1-NN com distância DTW; o de referência é o 1-NN com distância euclidiana. As acurácias obtidas foram calculadas sobre o particionamento canônico do repositório UCR. As acurácias esperadas foram calculadas por meio de validação *leave-one-out* nos exemplos de treinamento.

O gráfico *sharpshoot* é dividido em quatro regiões e cinco fronteiras associadas às relações de ganho esperado e obtido. Qualquer valor de ganho igual a 1 significa que foi verificada mesma acurácia para ambos os algoritmos na validação ou no teste. Valores de ganho superior a 1 significam que o algoritmo de interesse apresenta acurácia superior ao de referência. As regiões verdadeiro-positivo e verdadeiro-negativo (marcadas no gráfico como “VP” e “VN”) contêm os casos nos quais o ganho esperado foi confirmado pelo ganho obtido. As regiões verdadeiro-positivo e falso-negativo (“VP” e “FN”) contêm os casos nos quais a acurácia obtida do algoritmo de interesse foi superior à do algoritmo de referência—os mesmos pontos situados na região sombreada do gráfico de espalhamento). A região “FP” contém os casos considerados “piores”: os falso-positivos são aqueles nos quais se previu ganho, mas verificou-se perda.

Note que o gráfico *sharpshoot* não substitui, mas complementa o gráfico de espalhamento. Ele oferece uma visão do grau de confiabilidade que se pode depositar na escolha de um algoritmo em detrimento de outro. O gráfico mostrado na Figura 19 sugere que escolher a distância DTW em detrimento da distância euclidiana é em geral uma boa opção porque existem poucos casos falsos—em particular, poucos casos de falso-positivo. Note, por outro lado, que a posição do ponto no gráfico não indica se utilizar um algoritmo ou outro levará a resultados “adequados”. Por exemplo, o conjunto Phoneme se expressa no gráfico *sharpshoot* como o ponto mais distante

da origem. Isso quer dizer apenas que, tanto na etapa de validação quanto na etapa de teste, verificou-se que o 1-NN foi bastante superior com a distância DTW. Entretanto, como se pode observar na Figura 19, ambos os algoritmos produzem classificadores com baixa acurácia.

3.7.2 Teste de hipótese

Embora sejam úteis como ferramenta de visualização do desempenho de algoritmos, existem algumas desvantagens inerentes aos gráficos de espalhamento e *sharpshoot*, assim como outras ferramentas (não discutidas nesta Tese) como tabelas de valores médios, gráficos de barras e *boxplots*. Primeiramente, as conclusões obtidas através desses gráficos são subjetivas. Além disso, a utilidade dessas ferramentas é diminuída quando se deseja comparar mais de dois algoritmos.

Testes de hipótese são frequentemente utilizados para promover uma comparação objetiva de diferentes algoritmos. No contexto de classificação, o formato geral de um teste de hipótese consiste nos seguintes passos:

1. Estabelecer a hipótese nula de que todos os indutores testados produzem classificadores idênticos com respeito a alguma medida de qualidade (*e.g.*, acurácia);
2. Definir um procedimento de teste e estabelecer uma estatística de teste (ou de prova) que tenha uma distribuição de probabilidades conhecida;
3. Estabelecer um grau de confiança e verificar a probabilidade de que, assumindo-se a hipótese nula como verdadeira, o valor obtido para a estatística de teste seja aleatório;
4. Se essa probabilidade for inferior a um valor crítico, rejeitar a hipótese nula;
5. Se necessário, conduzir um teste post-hoc para determinar os indutores que são consistentemente superiores aos demais testados.

Vários testes de hipótese são adequados para comparar indutores (DEMSAR, 2006; DIETTERICH, 1998). Entretanto, nem todos são igualmente adequados para os mesmos cenários. Em particular, alguns testes assumem propriedades das distribuições das populações—no caso dos experimentos apresentados nesta Tese, as populações são os valores de acurácia obtida para cada classificador. Esses testes são denominados **paramétricos**. Além disso, muitos testes são adequados apenas para comparar dois classificadores, sendo necessário efetuar correções para comparação de múltiplos classificadores. Um exemplo bastante relevante é o teste t de Student. Sob a premissa de que a população segue uma distribuição normal, obtém-se uma estatística de prova com base nos escores padrões das observações. Com base nessa estatística de prova, encontra-se um valor crítico. Um erro comum é utilizar o teste t de Student para dados que não seguem distribuição normal. Outro erro comum é empregar vários testes t de Student sobre

pares de algoritmos e tentar propor uma conclusão sobre o desempenho de todos os algoritmos conjuntamente. Como as conclusões obtidas por cada teste possuem uma probabilidade de serem incorretas, esse tipo de repetições de testes pareados gera uma propagação de incerteza que pode levar a conclusões erradas.

Para populações razoavelmente grandes, é comum supor que os valores observados seguem uma distribuição normal ou aproximadamente normal. A partir de $N = 30$ exemplos, pode-se utilizar também testes de normalidade que verificam se esses exemplos seguem uma distribuição normal (DEMSAR, 2006). Não obstante, optou-se neste trabalho por utilizar testes de hipótese não paramétricos. A única desvantagem é que testes não paramétricos normalmente possuem menor poder do teste em comparação aos testes paramétricos. Isto é, testes paramétricos normalmente exigem menos “evidência” (dados) para rejeitar a hipótese nula.

Foram escolhidos dois testes não paramétricos para a avaliação experimental neste trabalho. O teste de Wilcoxon foi empregado para comparação de dois indutores. Para comparação de múltiplos indutores, foi utilizado o teste de Friedman com teste post-hoc de Nemenyi.

O teste de Wilcoxon ou, mais especificamente, o teste com sinal de *ranks* pareados de Wilcoxon, é um teste de hipótese não paramétrico para comparar dois algoritmos. Sejam $x_1, x_2, \dots, x_N$ valores de acurácia associados a um indutor I_x e $y_1, y_2, \dots, y_N$ valores de acurácia associados a um indutor I_y , com a restrição de que cada par (x_i, y_i) foi produzido com base nos mesmos exemplos de treinamento e teste. O primeiro passo do teste de Wilcoxon consiste em calcular as diferenças $d_i = x_i - y_i$. Em seguida, essas diferenças são ordenadas de acordo com seus valores absolutos e um *ranking* é produzido. Diferenças iguais ($|d_i| = |d_j|$) recebem a média dos *ranks*.

Um exemplo desse procedimento é dado na Tabela 4. O indutor I_x é o 1-NN com distância DTW e o indutor I_y é o 1-NN com distância euclidiana. Os valores mostrados nas colunas DTW e EUC são acurácia dos classificadores produzidos no particionamento “canônico” de alguns conjuntos de dados do repositório UCR.

Tabela 4 – Exemplo de *ranking* no teste de Wilcoxon

Conjunto	DTW	EUC	Diferença	Rank
ArrowHead	0,8000	0,8000	0,0000	1,5
ChlorineConcentration	0,6500	0,6500	0,0000	1,5
Beef	0,6670	0,6667	+0,0003	3,0
Adiac	0,6090	0,6113	−0,0023	4,0
CinCECGtorso	0,9300	0,8971	+0,0329	5,0
Car	0,7670	0,7333	+0,0337	6,0
BeetleFly	0,7000	0,7500	−0,0500	7,0
50Words	0,7580	0,6308	+0,1272	8,0
CBF	0,9960	0,8522	+0,1438	9,0
BirdChicken	0,7000	0,5500	+0,1500	10,0

No passo seguinte, os *ranks* associados às diferenças positivas são somados em R^+ e os *ranks* associados às diferenças negativas são somados em R^- . *Ranks* associados a casos

$x_i = y_i$ são distribuídos entre os dois algoritmos e, se houver um número ímpar desses casos, desconsidera-se um deles.

Para o exemplo da Tabela 4, as somas dos *ranks* são:

$$\begin{aligned} R^+ &= 1,5 + 3 + 5 + 6 + 8 + 9 + 10 = 42,5 \\ R^- &= 1,5 + 4 + 7 = 12,5 \end{aligned}$$

Seja $T = \min(R^+, R^-)$ e a estatística de prova z (Equação 3.23). Se o número de conjunto de dados for superior a 30, então essa estatística de prova z aproxima a distribuição normal padrão $\mathcal{N}(0, 1)$. Nesse caso, a hipótese nula de que os algoritmos são idênticos pode ser rejeitada, com confiança $\alpha = 0,05$, quando $z < -1,96$. Para um número menor de conjunto de dados, segue-se uma tabela de valores críticos sobre T (DEMSAR, 2006) e a hipótese nula é rejeitada se T for igual ou menor que o valor crítico da tabela.

$$z = \frac{T - \frac{1}{4}N(N+1)}{\sqrt{\frac{1}{24}N(N+1)(2N+1)}} \quad (3.23)$$

O valor crítico para $N = 10$ conjuntos de dados, com $\alpha = 0,05$, é $p = 8$. Como o resultado obtido foi $T = 12,5 > 8$, a hipótese nula de que os indutores são idênticos não pode ser rejeitada. Isso não significa, entretanto, que a hipótese nula deve ser aceita. Um teste de hipótese é sempre um teste de contradição, no qual a hipótese nula é assumida e, se o resultado obtido for “extremo”, então a hipótese nula pode ser rejeitada com certo grau de confiança. De fato, como os gráficos mostrados na seção anterior insinuam e como já foi evidenciado várias vezes pela comunidade científica, o classificador 1-NN é, no caso geral, mais eficaz quando se utiliza a distância DTW do que quando se utiliza a distância euclidiana.

No exemplo apresentado nesta seção, a quantidade de *evidência* fornecida ao teste de hipótese foi insuficiente para alcançar qualquer conclusão. Se repetirmos o mesmo experimento utilizando todos os conjuntos de dados do repositório UCR, obtemos $R^+ = 3.188,5$ e $R^- = 466,5$. Como o repositório UCR possui 85 conjuntos de dados, a estatística de prova z aproxima a normal padrão. Neste caso, a estatística de prova é $z = -5,85$, o que nos permite rejeitar a hipótese nula com $\alpha = 0,05$ e declarar que o indutor associado à maior soma de *ranks* pode ser considerado superior com 95% de confiança.

O teste de Friedman é um teste não paramétrico semelhante ao ANOVA (DEMSAR, 2006). Ele é adequado quando se deseja comparar múltiplos indutores sobre diversos conjunto de dados e também é baseado em *ranks*. A hipótese nula do teste de Friedman é simplesmente a suposição de que todos os indutores produzem classificadores idênticos.

Suponha que se deseja comparar k indutores sobre N conjuntos de dados. Para cada conjunto de dados, deve-se primeiramente construir um *ranking* de k valores de acurácia,

como exemplificado na Tabela 5. Empates são corrigidos com a média dos *ranks*. Em seguida, para cada indutor calcula-se o seu *rank* médio R_j , como também mostrados na Tabela 5. Os valores utilizados neste exemplo são um sub-conjunto dos resultados apresentados na Tabela 2 (pág. 60).

Tabela 5 – Exemplo de *ranks* em um teste de Friedman. Os valores entre parênteses são os *ranks* associados a cada indutor em cada conjunto de dados.

Conjunto	Tipo de normalização					
	Z-normalização		Intervalo [0, 1]		Histograma	
Beef	0,6667	(1,5)	0,6333	(3)	0,6667	(1,5)
BeetleFly	0,7500	(3)	0,8000	(2)	0,8500	(1)
BirdChicken	0,5500	(3)	0,6500	(2)	0,7000	(1)
Fish	0,7829	(1,5)	0,7771	(3)	0,7829	(1,5)
HAR	0,8809	(2)	0,8775	(3)	0,8812	(1)
OliveOil	0,8333	(2,5)	0,8333	(2,5)	0,8667	(1)
<i>Ranks</i> médios	$R_1 = 2,25$		$R_2 = 2,58$		$R_3 = 1,17$	

Para um número suficientemente grande de conjuntos de dados e de indutores (Demsar (2006) sugere $N > 10$ e $k > 5$), a estatística de prova χ_F^2 que segue uma distribuição χ^2 com $(k - 1)$ graus de liberdade. Entretanto, essa estatística é excessivamente conservadora. Deriva-se dessa estatística a estatística de teste F_F segue uma distribuição-F com $(k - 1)$ e $(k - 1)(N - 1)$ graus de liberdade.

$$\chi_F^2 = \frac{12N}{k(k+1)} \left[\sum_{j=1}^N R_j^2 - \frac{k(k+1)^2}{4} \right] \quad (3.24)$$

$$F_F = \frac{(N-1)\chi_F^2}{N(k-1) - \chi_F^2} \quad (3.25)$$

O valor crítico para F_F deve ser consultado de acordo com a distribuição F para os graus de liberdade definidos por N e k . Caso o valor da estatística de teste F_F seja superior ao valor crítico, então a hipótese nula de que os indutores produzem classificadores idênticos pode ser rejeitada.

Se for possível rejeitar a hipótese nula, pode-se realizar um teste post-hoc de Nemenyi para determinar quais indutores diferem com significância estatística. A distância crítica entre *ranks* para se declarar que os indutores são distintos, com um certo valor de confiança α , é dada por

$$CD_{\alpha,k} = q_{\alpha,k} \sqrt{\frac{k(k+1)}{6N}}, \quad (3.26)$$

no qual o parâmetro $q_{\alpha,k}$ pode ser consultado na Tabela 6.

Para o exemplo da Tabela 5, a estatística F_F é uma aproximação muito fraca da distribuição-F devido ao baixo número de conjunto de dados, portanto o exercício seguinte é apenas ilustrativo (nos experimentos desta Tese, serão utilizados dezenas de conjuntos de dados).

Tabela 6 – Valores críticos para o teste de Nemenyi (DEMSAR, 2006).

α	Número de indutores (k)								
	2	3	4	5	6	7	8	9	10
0,05	1,960	2,343	2,569	2,728	2,850	2,949	3,031	3,102	3,164
0,10	1,645	2,052	2,291	2,459	2,589	2,693	2,780	2,855	2,920

O valor crítico da distribuição-F para 2 e 10 graus de liberdade⁷ é 4,10. A estatística de prova é $F_F = 6,08$. Como o valor da estatística de prova é *superior* ao valor crítico, então a hipótese nula de que os três indutores são idênticos deve ser rejeitada.

A distância crítica para $k = 3$ classificadores, com $\alpha = 0,05$, é $CD = 1,35$. Isso significa que os indutores cuja diferença de *rank* médio for superior a 1,35 devem ser considerados distintos. A única diferença de *ranks* superior à distância crítica, para o exemplo da Tabela 5, ocorre em $|R_3 - R_2| = 1,41$. Dessa forma, o teste só consegue denunciar diferença estatisticamente significativa entre os métodos de normalização como histograma e normalização para o intervalo $[0, 1]$. O método associado a menor *rank* médio é considerado superior.

Uma forma mais conveniente de se ilustrar resultados desse tipo de teste de hipótese é o gráfico de distância crítica (Figura 21) ou, mais simplesmente, gráfico CD (*critical distance*). O eixo único do gráfico indica a escala de *ranks*. Cada algoritmo comparado é associado a uma posição nesse eixo que equivale ao seu *rank* médio. A linha grossa tracejada na parte superior do gráfico sob o termo “CD” indica a distância necessária entre dois ou mais *ranks* no eixo para que os algoritmos sejam considerados distintos. Os algoritmos que não diferem significativamente são conectados por linhas grossas na parte inferior do gráfico.

Figura 21 – Gráfico de distância crítica para o exemplo da Tabela 5.

Note que este se trata apenas de um exemplo numérico do teste de Friedman e do teste post-hoc de Nemenyi. Primeiramente, o número de conjuntos de dados utilizados neste experimento é muito reduzido e a aproximação da estatística de prova F_F para a distribuição-F que realizamos é, no mínimo, questionável. Além disso, o teste foi realizado sobre um subgrupo de conjuntos de dados selecionados especificamente para apresentar os resultados desejados. Resultados experimentais verdadeiros desses três mesmos classificadores sobre todos os conjuntos de dados do repositório UCR serão apresentados na Seção 3.9.3 desta Tese.

⁷ $(k - 1) = 2$; e $(k - 1)(N - 1) = 2 \cdot 5 = 10$.

Com isso, finalizamos a revisão das ferramentas de teste empregadas nesta Tese. Note que nenhuma das quatro ferramentas apresentadas nesta seção pode ser utilizada como prova definitiva de qualquer conclusão, entretanto cada uma delas fornece evidência que pode ser utilizada para defender uma hipótese. No caso dos gráficos, quanto mais extrema a distribuições dos pontos, mais confiança pode ser depositada na conclusão de que os resultados obtidos não podem ser explicados por variações aleatórias dos dados ou da realização dos experimentos. Similarmente, quanto mais extremos forem os valores das estatísticas de teste, mais confiança pode ser depositada na conclusão de que os resultados obtidos não serão contraditos por experimentos futuros.

3.8 Ajuste de parâmetros

Um caso particular de teste de indutores ocorre quando um algoritmo de indução ou classificação possui parâmetros que devem ser estabelecidos para cada aplicação. Em alguns casos, existem parâmetros que são tipicamente associados a valores “típicos” para certos domínios de aplicação. Por exemplo, para classificação de séries temporais, é comum utilizar-se o parâmetro $k = 1$ para determinar o tamanho da vizinhança no classificador k -NN. Entretanto, como não se assume nenhuma aplicação em particular neste trabalho, promove-se ajuste de parâmetros para cada algoritmo por meio de validação cruzada *leave-one-out*. Essa escolha foi justificada por avaliação experimental apresentada na Seção 3.9.1.

3.9 Avaliação experimental

Ao longo deste capítulo, foram levantadas algumas hipóteses que podem ser importantes para as argumentações desta Tese. Algumas dessas hipóteses já foram suficientemente verificadas por outros autores. Por exemplo, em trabalhos de classificação de séries temporais, é tipicamente *aceito* que as observações das séries temporais não podem ser utilizadas como atributos de forma eficiente—o que justifica o imenso esforço da comunidade de aprendizado de máquina para extrair atributos e definir estratégias de classificação por instância de séries temporais. Outras hipóteses, mais específicas ao trabalho realizado no desenvolvimento desta Tese, foram verificadas experimentalmente. Além disso, foram realizados experimentos para justificar a escolha de uma estratégia de reamostragem no caso de ajuste de parâmetros. Os resultados e as conclusões alcançados são apresentados nas subseções a seguir.

3.9.1 Comparação de métodos de validação

Existem diferentes métodos de reamostragem que podem ser utilizados sobre os exemplos de treinamento quando se deseja, por exemplo, fazer ajuste de parâmetros ou construir gráficos *sharpshoot*. Observa-se, na literatura científica, uma preferência por validação cruzada *leave-*

one-out (SHOKOOHI-YEKTA *et al.*, 2015) ou 10-fold CV (KATE, 2016; JANYALIKIT *et al.*, 2016). Uma vantagem do *leave-one-out* é que se trata de um procedimento determinístico, enquanto o 10-fold CV depende de um particionamento aleatório em *folds*.

Para definir a estratégia de reamostragem a ser utilizada na estimativa da acurácia esperada, foi realizado um experimento comparativo do viés e da variância dos métodos *leave-one-out*, 10-fold CV, *bootstrap* e 5×2 CV. O viés foi determinado através de um escore obtido pela diferença entre a acurácia esperada e a acurácia obtida. Isto é, para cada conjunto de dados, foi calculado o escore $\text{acc}_v = \text{acc}_e - \text{acc}_a$, no qual os subscritos indicam, respectivamente, escore de viés, acurácia esperada e acurácia obtida. Um escore negativo sugere que o método de reamostragem possui um viés pessimista, ao passo que um escore positivo indica que o método possui um viés otimista.

Para estimar a variância dos métodos, esse procedimento foi repetido 30 vezes com reamostragens aleatórias “estratificadas” dos particionamentos canônicos do repositório UCR. Isto é, dado um conjunto de treinamento T e um conjunto de testes V , primeiramente obteve-se um conjunto de todas as instâncias do problema $F = T \cup V$ e então amostrou-se novos conjuntos de treinamento e testes T' e V' , de modo que a distribuição das classes em T' era idêntica à de T e a distribuição das classes em V' era idêntica à de V . O método de reamostragem foi aplicado em T' , do qual se derivou um único valor de acurácia esperada. O conjunto V' foi utilizado para se encontrar a acurácia obtida. Esse procedimento foi repetido 30 vezes para cada conjunto de dados. Os resultados foram representados como *boxplots* e estudados. Alguns casos são apresentados na Figura 22. A coleção completa de *boxplots* para todos os conjuntos do repositório UCR é apresentada na Seção A.1 (pág. 215).

O objetivo deste estudo era verificar se havia alguma diferença substancial e consistente do viés e da variância dos estimadores conforme os dados eram reamostrados. Os gráficos devem ser lidos da seguinte forma: as partes superiores e inferiores das caixas representam o terceiro e o primeiro quartis dos escores de viés nos 30 casos de reamostragem; a linha central (vermelha, se este texto estiver disponível em cores) representa a mediana. Os bigodes indicam os valores máximo e mínimo, na ausência de *outliers*, os quais são indicados como pontos rotulados com o símbolo “+”. Quanto maior a distância entre os quartis, maior a variância verificada nos testes. Quanto mais próximos os valores estiverem de zero, menor o viés do estimador.

Os casos mostrados na Figura 22 são representativos do que se verificou nos conjuntos restantes: o 10-fold CV tende a produzir estimadores ligeiramente mais otimistas e, na maioria dos casos, com menor viés que validação cruzada por 5×2 CV e *bootstrap*. Entretanto, a variância do 10-fold CV é próxima do *leave-one-out* na maioria dos casos. Os menores casos de variância foram observados quando se utilizou reamostragem por 5×2 CV ou *bootstrap*, ao custo de um viés bastante elevado em comparação com 10-fold CV e *leave-one-out*.

Como esse estudo preliminar não revelou diferença substancial de viés ou de variância entre os métodos de reamostragem 10-fold CV e *leave-one-out*, optou-se por não prosseguir

Figura 22 – Scores de viés para quatro conjuntos do repositório UCR.

com um estudo mais aprofundado e utilizar um desses dois métodos, que são os mais comuns na literatura. A escolha pelo *leave-one-out* em detrimento do 10-fold CV se deveu principalmente ao fato de que o primeiro se trata de um método determinístico.

3.9.2 Número de vizinhos no k -NN

Um parâmetro que é tipicamente ignorado na literatura de classificação de séries temporais é o tamanho da vizinhança do classificador k -NN. Um dos trabalhos mais antigos que utiliza um algoritmo de vizinhos mais próximos na classificação de séries temporais é o de (GEURTS, 2001). Muitos outros trabalhos relacionados a mineração de dados temporais desde o início dos anos 2000 também vêm focando exclusivamente no classificador 1-NN (KEOGH; KASETTY, 2003). Embora certamente haja publicações nas quais o tamanho da vizinhança do classificador k -NN tenha sido explorado (CHAOVALITWONGSE; FAN; SACHDEO, 2007), os trabalhos mais semelhantes a este projeto promovem classificação por similaridade através do classificador 1-NN (BAGNALL *et al.*, 2015).

Um dos experimentos realizados durante a execução deste projeto consistiu em promover uma comparação de diferentes parâmetros de k para os conjuntos do repositório UCR. Nesse experimento, foi utilizado o particionamento canônico e os classificadores foram comparados com base nos valores de acurácia obtida. Para cada conjunto de dados, foi realizada classificação dos exemplos de teste utilizando k -NN com $k = 1, 2, \dots, 16$. Para os casos em que $k > 1$, foi considerada uma única modificação do Algoritmo 3.3: ao classificar um novo exemplo, se a

moda da classe na vizinhança desse exemplo puder se atribuída a mais de um rótulo, então o algoritmo retorna um conjunto contendo todos esses rótulos. Nesse caso, a contribuição desse exemplo para o cálculo da acurácia é ponderada inversamente pelo número de valores nesse conjunto e a acurácia obtida é um valor médio caso o critério de desempate fosse a escolha aleatória de um dos rótulos presentes na moda.

Os resultados completos são omitidos por serem excessivamente longos, mas podem ser consultados em (GIUSTI, 2017). Entretanto, os resultados são facilmente reproduzíveis e mostram que o k -NN é consistentemente mais fiel ao conceito das classes para valores menores de k . A fim de promover uma comparação mais objetiva, foi conduzido um teste de hipótese de Friedman considerando a vizinhança do k -NN variando em um espaço de busca reduzido. Foram considerados valores do parâmetro k variando de $k = 1$ a $k = 9$. Após rejeitar a hipótese nula de que todos os classificadores são idênticos, foi produzido um gráfico CD de acordo com o teste post-hoc de Nemenyi (Figura 23). O teste denuncia a existência de um grupo de valores do parâmetro k que produzem classificadores significativamente superiores quando o tamanho da vizinhança é reduzido. Porém, não é possível diferenciar dentre esses valores devido ao elevado número de classificadores envolvidos no teste. Seria necessário mais evidência (*i.e.*, mais conjuntos de dados) para que se pudesse afirmar diferença estatisticamente significativa com relação a esses valores do parâmetro k .

Figura 23 – Gráfico CD comparando o classificador k -NN com vizinhanças de $k = 1$ a $k = 9$.

Assim, foi realizado um novo teste de Friedman seguido de um teste post-hoc de Nemenyi considerando apenas os valores de vizinhança do primeiro grupo ($k \in \{1, 3, 4, 5\}$). Um gráfico de distância crítica para esse experimento é apresentado na Figura 24. Esse segundo teste denunciou que os classificadores 1-NN e 3-NN são superiores, com diferença estatisticamente significativa, aos classificadores 4-NN e 5-NN, sem distinção entre os dois primeiros.

Figura 24 – Gráfico CD comparando o classificador k -NN com vizinhanças de $k \in \{1, 3, 4, 5\}$.

Esses resultados experimentais sugerem que o esforço computacional em ajustar o parâmetro k com valores de vizinhança variando de $k = 1$ a $k = 16$ não seria recompensado com

resultados melhores de classificação. Portanto, seguiu-se neste trabalho a tendência percebida na literatura de se favorecer o classificador 1-NN para a classificação de séries temporais.

3.9.3 Métodos de normalização

Como mencionado anteriormente, os trabalhos relacionados a esta Tese parecem favorecer a z -normalização (ou normalização em escores padrão) em detrimento de outros métodos, como a normalização para o intervalo $[0, 1]$ ou a normalização em histogramas—que portanto são doravante referidos como métodos “alternativos” de normalização. Na Seção 2.6.2, foram apresentados resultados de um estudo preliminar no qual foi utilizado um pequeno número de conjuntos de dados para verificar a acurácia do classificador 1-NN sem normalização, com z -normalização e com métodos alternativos de normalização. Foram selecionados alguns conjuntos de séries temporais do repositório UCI, bem como alguns conjuntos do repositório UCR que, à época, eram disponibilizados sem normalização prévia.

Os resultados daquele estudo foram motivadores para se testar uma hipótese semelhante com todos os conjuntos do repositório UCR. Como o repositório atualmente disponibiliza todos os dados normalizados em escores padrão, os testes conduzidos tiveram como objetivo comparar os métodos alternativos de normalização com relação à z -normalização. Os resultados completos são apresentados no apêndice desta Tese (Seção A.2, pág. 219). Na Figura 25, apresenta-se um sumário na forma de gráficos de espalhamento.

Figura 25 – Gráficos de espalhamento comparando métodos de normalização.

Com base nos resultados obtidos, foi conduzido um teste de hipótese Friedman. O valor da estatística-F encontrada para o experimento foi 2,21 e o valor crítico para 3 algoritmos e 85 conjuntos com $\alpha = 0,05$ é 3,05. Como o valor de prova é inferior ao valor crítico, o teste não foi capaz de rejeitar a hipótese nula de que os três métodos de normalização produzem algoritmos idênticos. Isso não significa que os métodos sejam idênticos, mas que as evidências reunidas não permitem afirmar que um método de normalização produz classificadores consistentemente melhores que outros utilizando 1-NN com distância euclidiana.

A próxima hipótese levantada é a existência de casos particulares de domínios de aplicação nos quais um método seja preferível aos demais. Como se pode observar na Figura 25 e também na Seção A.2, houve alguns casos nos quais o desempenho de um método alternativo se mostrou bastante superior ao da z -normalização. Embora na maioria dos casos a diferença seja relativamente pequena, o uso de métodos alternativos de normalização ainda pode ser preferível se formos capaz de prever, através de reamostragem dos exemplos de treinamento, qual método será superior. Assim, o mesmo experimento foi repetido com validação *leave-one-out* e os resultados são apresentados como gráficos *sharpshoot* na Figura 26.

Figura 26 – Gráficos *sharpshoot* comparando métodos de normalização. Em ambos os casos, o ganho é definido como a acurácia do método de interesse sobre a acurácia da normalização em escores padrões.

As “áreas de interesse” do gráfico *sharpshoot* são aquelas sombreadas, nas quais se verifica os casos de verdadeiro-positivo e verdadeiro-negativo. Para ambos os métodos alternativos de normalização, verificou-se uma ocorrência significativa de falsos-negativos e, particularmente, de falsos-positivos. Isso sugere que a tentativa de selecionar um método de normalização com reamostragem dos dados de treinamento não é adequada.

Com base em todos esses resultados, a conclusão mais razoável é que as pequenas variações observadas nos experimentos são resultados aleatórios que podem ser explicada pela variância naturalmente associada ao procedimento de reamostragem. A decisão tomada foi, portanto, de utilizar a estratégia de z -normalização como “padrão” de normalização nos experimentos desta Tese, exceto para casos de funções de distância particulares, como será mostrado na Seção 4.5.1.

3.10 Considerações finais

Neste capítulo foi realizada uma revisão bibliográfica dos principais conceitos de aprendizado de máquina e suas contribuições para a mineração de dados, em especial para promover

classificação de séries temporais. Foram revisados também os métodos de validação de classificadores e algoritmos de indução de classificadores. No restante desta Tese, os métodos de validação, tanto os visuais (grafos de espalhamento e *sharpshoot*) quanto os testes de hipóteses, serão empregados extensamente para avaliar os resultados experimentais produzidos durante a execução desta pesquisa.

Na Seção 3.9 e em suas subseções, foram apresentados resultados experimentais de hipóteses relacionadas à forma “ideal” de se avaliar classificadores, ao ajuste de parâmetros do classificador 1-NN e ao emprego de métodos de normalização. Esses resultados embasam decisões importantes que definem o escopo dos experimentos que serão apresentados nos próximos capítulos desta Tese:

- A escolha de validação cruzada *leave-one-out* como estratégia para ajuste de parâmetros e também para a escolha de classificadores base na composição de *ensembles* de classificadores (Capítulo 6);
- A escolha do classificador 1-NN em detrimento do k -NN utilizando um tamanho de vizinhança $k \neq 1$;
- A escolha da z -normalização em detrimento de outros métodos de normalização como “padrão” dos experimentos desta Tese. Isso não significa, entretanto, que outros métodos de normalização foram descartados. Esses métodos foram utilizados excepcionalmente com funções de distâncias que exigem que as observações sejam não negativas (normalização em $[0, 1]$) ou que as observações sejam não negativas e a área sob a curva da série seja unitária (normalização em histogramas).

Uma decisão não justificada por experimentos neste capítulo foi a escolha de um classificador baseado na regra dos vizinhos mais próximos em detrimento de outros modelos, como o classificador ingênuo de Bayes, árvores de decisão, máquinas de vetores de suporte (Seção 5.7) ou redes neurais (GILES; LAWRENCE; TSOI, 2001). Ocorre que, na literatura científica moderna, as práticas dominantes de classificação de séries temporais são aquelas baseadas na extração de atributos de maior ordem ou em instâncias. Nesta Tese, a escolha por um classificador baseado em instâncias foi simplesmente parte da definição do escopo desta pesquisa.

Mesmo assim, existem diversos classificadores baseados em instâncias, além do k -NN. A escolha do 1-NN é justificada principalmente pela sua prevalência na literatura científica. O “estado-da-arte”, à época em que este trabalho foi iniciado, era o classificador 1-NN aliado à distância DTW. Nos últimos quatro anos, diversos trabalhos têm sido publicados com alegações de terem superado esse modelo de classificação. Alguns autores têm procurado avançar essa fronteira de pesquisa com redes neurais (ZHENG *et al.*, 2014), máquinas de vetores de suporte (KATE, 2016) ou *ensembles* complexos de classificadores (BAGNALL *et al.*, 2015). O escopo definido para esta pesquisa foi buscar formas de associar diversidade de representações e

de funções de distância ao classificador 1-NN para tentar avançar o estado-da-arte. Não obstante, a proposta de se associar diversidade de representações a outros modelos de classificação foi considerada neste trabalho, como será discutido no Capítulo [5](#).

DISTÂNCIAS ENTRE SÉRIES

4.1 Considerações iniciais

OS MÉTODOS DE CLASSIFICAÇÃO POR SIMILARIDADE dependem fortemente de um mecanismo capaz de expressar a semelhança entre objetos associados a diferentes conceitos. Uma função de distância ou similaridade é fundamental, por exemplo, para que o classificador 1-NN seja capaz de distinguir objetos pertencentes a classes distintas.

Séries temporais podem ser comparadas sob uma infinidade de aspectos com respeito a características discriminantes como complexidade, “forma” da curva, presença ou ausência de tendências e ciclos, invariâncias, dentre outras. Portanto, é importante que se possa estudar e comparar várias funções, compreender as diferenças entre elas e observar suas limitações. Isso permite identificar características gerais dos dados que sugerem se alguma função de distância é potencialmente mais adequada que outras.

Muitas funções de distância foram propostas para diferentes tarefas de análise de dados temporais. Neste capítulo, algumas das distâncias identificadas como as mais importantes para a realização deste trabalho são apresentadas e estudadas. Na Seção 4.7 e em suas subseções, são apresentados resultados experimentais mostrando a diferença de classificadores associados às várias funções de distância nos conjuntos de dados do repositório UCR.

4.2 Atributos desejáveis de funções de distância

Dado um conjunto X e $\{x, y, z\} \subset X$, diz-se que uma **métrica** ou uma **função de distância** é um mapeamento $f : X \times X \mapsto \mathbb{R}$ tal que as seguintes relações são verdadeiras:

1. $f(x, y) \geq 0$ (não negatividade)
2. $f(x, y) = 0 \Leftrightarrow x = y$ (identidade dos indiscerníveis)

3. $f(x, y) = f(y, x)$ (simetria)
4. $f(x, z) \leq f(x, y) + f(y, z)$ (desigualdade triangular)

Uma função de distância entre séries temporais é um mapeamento $d : \mathbb{T} \times \mathbb{T} \mapsto \mathbb{R}$ que produz um valor de escore inversamente correlacionado com algum conceito de semelhança ou proximidade. Por exemplo, o escore da distância euclidiana é obtido de um alinhamento de duas séries no eixo temporal e trata-se de um exemplo de “similaridade no tempo”—cada “instante” no qual as séries são diferentes contribuem positivamente para o escore total.

As propriedades de métricas podem ser importantes em diversas circunstâncias na análise de séries temporais. A não negatividade é assumida quando se estabelece conceitos de vizinhança. A identidade dos indiscerníveis assegura que uma distância $f(\cdot, \cdot) = 0$ só pode ocorrer (e sempre ocorre) quando os objetos apresentarem propriedades idênticas—em outras palavras, quando todas as observações forem idênticas. A simetria, para o classificador k -NN, garante que o escore de distância entre as séries independe de qual é a instância de prova e qual é parte do conjunto de treinamento. E a desigualdade triangular estabelece relações espaciais que podem ser utilizadas, por exemplo, na indexação de séries temporais (YE *et al.*, 2009).

Por outro lado, muitas vezes opta-se por utilizar uma função não métrica que seja mais adequada para alguma aplicação. Um exemplo é a distância DTW, que não proporciona a desigualdade triangular e relaxa a propriedade 2 para o seguinte:

2. $f(x, x) = 0$

Isto é, a distância DTW entre uma série para si própria é sempre zero, mas não necessariamente duas séries temporais cuja distância DTW é nula são idênticas. Isso significa que a distância DTW, em sua formulação original, não pode ser considerada uma métrica. Apesar disso, como será mostrado ao longo deste capítulo, esse é um fato normalmente justificado pelos benefícios fornecidos pela DTW com respeito à invariância a *warping* local.

Outras funções podem ser ainda mais irrestritas com respeito às condições de métrica. A distância de Kullback-Leibler (Equação 4.17) é um exemplo de distância assimétrica. Entretanto, normalmente não se admite distâncias que desrespeitam a propriedade da não negatividade. Tipicamente, considera-se que uma função de distância que escores negativos está mal-formulada ou é inadequada para o domínio no qual está sendo empregada, como será mostrado adiante neste capítulo.

Estritamente, um mapeamento que não respeita todas as propriedades levantadas no início desta seção não pode ser considerado uma função de distância, “apenas” uma função de dissimilaridade. Mas a literatura de séries temporais usualmente emprega o termo “distância” para referir-se a qualquer função de comparação que produz um escore de dissimilaridade entre séries temporais, reservando o termo “métrica” para distâncias que respeitam a desigualdade

triangular. Nesta Tese, exceto quando o contexto exigir, o termo “distância” será empregado indiscriminadamente para métricas e funções de dissimilaridade.

4.3 Classes de funções de distância

Funções de distância para séries temporais podem ser agrupadas da seguinte forma:

1. As métricas da família Minkowski (Seção 4.4);
2. As demais distâncias não elásticas (Seção 4.5);
3. As distâncias elásticas (Seção 4.6).

As distâncias **não elásticas**, também conhecidas como *lockstep measures*, são aquelas que comparam duas séries de maneira síncrona com relação à ordem das observações. Isto é, a distância não elástica promove um alinhamento entre cada t -ésima observação de uma série com a t -ésima observação da outra—como exemplificado na Figura 5 na pág. 38. Distâncias não elásticas exigem que ambas as séries possuam o mesmo número de observações e não admitem omissões de dados sem adaptações. Todas as métricas da família Minkowski, dentre as quais destaca-se a distância euclidiana, são não elásticas.

As distâncias **elásticas**, por sua vez, são aquelas que comparam duas séries por meio de um alinhamento assíncrono de suas observações. Distâncias elásticas podem alinhar uma observação a múltiplas observações de outra série ou desconsiderar segmentos das séries no cálculo da distância. Tais medidas tipicamente permitem comparação de séries com dimensionalidades distintas e frequentemente admitem omissões de dados.

4.4 Métricas da família Minkowski

A distância **euclidiana** é uma das mais simples e utilizadas em análise de séries temporais. Considerando-se séries temporais como vetores em um espaço cujas dimensões são determinadas pelos instantes de amostragem $t = 1, 2, \dots, n$, a distância euclidiana surge naturalmente a partir da norma L_2 e é definida pela Equação 4.1.

$$d_{\text{Euc}}(S, Z) = \sqrt{\sum_{t=1}^n |s[t] - z[t]|^2} \quad (4.1)$$

A distância euclidiana é o caso mais notável de distância não elástica e é também um exemplo (por definição) de métrica da **família Minkowski**. Métricas de Minkowski são

generalizações do conceito “tradicional” de distância empregado na geometria euclidiana. São também denominadas normas L_p e são expressadas conforme a Equação 4.2.

$$d_{\text{Mink}}(S, Z) = \left(\sum_{t=1}^n |s[t] - z[t]|^p \right)^{\frac{1}{p}} \quad (4.2)$$

Os exemplos mais importantes dessa métrica ocorrem em $p = 1$, $p = 2$ e $p \rightarrow \infty$. A distância **Manhattan** ocorre quando $p = 1$ e, no plano, equivale à soma das “linhas retas” que separam dois pontos em cada dimensão. Por esse motivo, ela é também conhecida como distância **city block**, pois representa a distância percorrida quando se caminha pelas quadras de uma cidade. Generalizada para n dimensões, é dada pela Equação 4.3.

$$d_{\text{Man}}(S, Z) = \sum_{t=1}^n |s[t] - z[t]| \quad (4.3)$$

Um caso “extremo” da família Minkowski ocorre quando p tende ao infinito. Esse caso particular da distância de Minkowski é conhecido como distância de **Chebyshev**. É dada pelo limite de (4.2) quando $p \rightarrow \infty$, o qual se resolve como na Equação 4.4.

$$d_{\text{Cheb}}(S, Z) = \max_{t=1}^n |s[t] - z[t]| \quad (4.4)$$

Existe literalmente uma infinidade de métricas pertencentes à família de Minkowski, determinadas pelo parâmetro p . O impacto desse parâmetro na concepção da função de distância é mostrado na Figura 27. Cada sub-figura representa o formato de um perímetro unitário obtido por um certo valor de p em duas dimensões. O caso comum é o da circunferência observada em $p = 2$. Conforme $p \rightarrow 0$, o espaço se contrai e a “contribuição” de um coeficiente é inversamente proporcional ao valor do segundo; já para $p \rightarrow \infty$, ocorre o oposto. É possível observar isso “acompanhando” a altura da curva unitária—o valor da coordenada em y —para diferentes valores de x no intervalo $[0, 1]$.

Generalizado para n dimensões, o efeito do parâmetro p é semelhante ao que se observa na figura. Quanto menor o valor de p , maior a “contribuição” relativa de pequenas divergências entre observações da série. É importante reforçar o uso da palavra “relativa”, pois a forma como esse parâmetro afeta a distância depende principalmente da dimensionalidade da série. No sentido inverso ($p \rightarrow \infty$), a “contribuição” individual de cada observação tende a ser menos expressiva. Uma ilustração desse fenômeno é dada para as séries na Figura 28-esquerda. A série $S = (0, 5, 4, 5, 4, 5)$ está representada com o marcador +; a série representada com o marcador * é $Z = (0, 3, 2, -3, -2, 6)$.

As curvas na Figura 28-direita representam as distâncias entre as subsequências nos intervalos 1..1, 1..2 e assim por diante. Isto é, em $t = 2$ mostra-se as distâncias entre a subsequência $s[1:3]$ e $z[1:3]$ —não a soma parcial dos fatores $(s[t] - z[t])^p$ para $t = 1..3$. A maior divergência

Figura 27 – Raio unitário em diferentes espaços L^p .

ocorre em $t = 4$. Note que, para $p = 0,5$, isso representa um grande “salto” do intervalo 1..3 para o intervalo 1..4. A divergência no instante $t = 5$ não é tão elevada, mas todavia representa um “salto” ainda mais expressivo para $p = 0,5$, pois as subsequências são mais longas e a contribuição relativa de cada par de observações é maior. Em contrapartida, para $p = 1$ nota-se que a todas as divergências contribuem de forma diretamente proporcional à diferença de “altura” entre as observações. Já para a distância Chebyshev, todas as subsequências que incluem $t = 4$ são determinadas pela divergência nesse ponto—a contribuição individual de todas as outras observações é nula.

A distância Minkowski é um mapeamento linear, o que significa que ela pode ser combinada com outras distâncias e o resultado ainda assim é uma métrica. Um exemplo relevante é obtido pela média aritmética da distância de Manhattan com a distância Chebyshev, apresentado

Figura 28 – Diferença de p na distância entre séries temporais.

na Equação 4.5.

$$d_{\text{AvgLp}} = \frac{\sum_{t=1}^n (|s[t] - z[t]|) + \max_{t=1}^n (|s[t] - z[t]|)}{2} \quad (4.5)$$

Em experimentos, a média das distâncias Manhattan e Chebyshev apresentou resultados competitivos com a distância euclidiana e em alguns casos superiores aos da DTW. O logaritmo da distância Manhattan produz outra métrica eficaz denominada Lorentzian—apresentada na próxima seção.

4.5 Outras medidas não elásticas

As distâncias de **Canberra** (Equação 4.6), **Lorentzian** (Equação 4.8) e de **Sørensen** (Equação 4.7) são medidas que derivam da distância de Manhattan. A distância de Canberra é uma normalização da distância de Manhattan na qual as divergências são mais substanciais para observações de baixa magnitude. A distância de Lorentzian obtém efeito semelhante, mas utiliza logaritmo para promover uma normalização não linear entre as observações. A distância de Sørensen é uma adaptação de um índice utilizado na ecologia para estimar a semelhança entre dois sítios com respeito a suas diversidades biológicas. Cha (2007) apresentou uma adaptação dessa distância para comparar funções de distribuição de probabilidade. A versão mostrada na Equação 4.7 emprega um módulo no denominador para lidar com observações negativas.

$$d_{\text{Can}}(S, Z) = \sum_{t=1}^n \frac{|s[t] - z[t]|}{|s[t]| + |z[t]|} \quad (4.6)$$

$$d_{\text{Sor}}(S, Z) = \frac{\sum_{t=1}^n |s[t] - z[t]|}{\sum_{t=1}^n (|s[t]| + |z[t]|)} \quad (4.7)$$

$$d_{\text{Lor}}(S, Z) = \sum_{t=1}^n \log(1 + |s[t] - z[t]|) \quad (4.8)$$

A distância **cosseno** (Equação 4.9) deriva do índice de similaridade cosseno, utilizado na mineração de textos como um índice de semelhança entre documentos. A distância cosseno considera apenas a orientação dos vetores de atributos (ou observações, no caso de séries temporais) dos objetos comparados, ignorando suas magnitudes. A distância cosseno pertence a uma família de distâncias denominada *produto interno*. Outra distância dessa família é a distância de **Jaccard**, derivada do coeficiente de correlação de Jaccard e apresentado na Equação 4.10.

$$d_{\text{Cos}}(S, Z) = 1 - \frac{\sum_{t=1}^n (s[t] - z[t])^2}{\|S\| \|Z\|} \quad (4.9)$$

$$d_{\text{Jac}}(S, Z) = \frac{\sum_{t=1}^n (s[t] - z[t])^2}{\sum_{t=1}^n (s[t] - z[t])^2 + s[t] \cdot z[t]} \quad (4.10)$$

O operador $\|\cdot\|$ é a *norma* (“tamanho”) de um vetor. A norma de uma série temporal pode ser calculada como na Equação 4.11. Experimentalmente, verifica-se que as distâncias cosseno e de Jaccard correlacionam com alta probabilidade com a distância euclidiana para séries z -normalizadas. Na prática, isso tem como efeito que o classificador 1-NN tende a produzir os mesmos resultados quando associado a quaisquer dessas medidas.

$$\|S\| = \sqrt{\sum_{t=1}^n s[t]^2} \quad (4.11)$$

Algumas funções de distância exploradas nesta Tese derivam da estatística de prova do teste χ^2 de Pearson, utilizado para testar a hipótese de que um conjunto de observações S não pode ter sido amostrado de uma variável aleatória $\mathbf{Z}$ cuja função de distribuição de probabilidades é conhecida. Sejam $E_1, E_2, \dots, E_n$ os possíveis resultados de $\mathbf{Z}$ e $H_S(E_i)$ um histograma de frequências de cada evento E_i na amostra S . Então, o coeficiente χ^2 de Pearson é dado pela Equação 4.12. Esse coeficiente se aproxima assintoticamente de uma distribuição χ^2 cujos parâmetros podem ser extrapolados de S . Caso o valor de χ^2 seja superior a um valor crítico associado a um valor de confiança, rejeita-se a hipótese nula de que H_S e $P(\mathbf{Z})$ são semelhantes.

$$\chi^2 = \sum_{i=1}^n \frac{(H_S(E_i) - P(\mathbf{Z} = E_i))^2}{P(\mathbf{Z} = E_i)} \quad (4.12)$$

Por abuso de notação, pode-se calcular o coeficiente χ^2 para uma série temporal S contra uma série temporal Z . Se as séries forem normalizadas como histogramas, então tem-se a garantia de que o coeficiente resultante segue alguma distribuição χ^2 . O problema de se comparar duas séries temporais se torna o problema de aproximar a “distribuição de probabilidades” Z com base na “distribuição de probabilidades” S .

Esse raciocínio permite derivar algumas distâncias que compõem a **família** χ^2 . A distância χ^2 **de Pearson** (Equação 4.13) é uma função assimétrica da distância entre S e Z quando S é “desconhecida” e Z é “conhecida”. A distância **de Neyman** χ^2 (Equação 4.14) é a relação simétrica da χ^2 de Pearson e difere desta com respeito à série que corresponderia à distribuição conhecida. As distâncias χ^2 **quadrada** (Equação 4.15) e **divergência** (Equação 4.16) são vari-

antes do coeficiente χ^2 de Pearson que empregam uma relação entre as duas distribuições para serem simétricas.

$$d_{\text{Pear}\chi^2}(S, Z) = \sum_{t=1}^n \frac{(s[t] - z[t])^2}{z[t]} \quad (4.13)$$

$$d_{\text{Ney}\chi^2}(S, Z) = \sum_{t=1}^n \frac{(s[t] - z[t])^2}{s[t]} \quad (4.14)$$

$$d_{\text{Sq}\chi^2}(S, Z) = \sum_{t=1}^n \frac{(s[t] - z[t])^2}{s[t] + z[t]} \quad (4.15)$$

$$d_{\text{Div}}(S, Z) = 2 \sum_{t=1}^n \frac{(s[t] - z[t])^2}{(s[t] + z[t])^2} \quad (4.16)$$

Outras distâncias empregadas nesta Tese advêm do conceito de entropia de Shannon. A distância de **Kullback-Leibler** (KL, Equação 4.17) é um coeficiente de dissimilaridade que representa a quantidade de informação “perdida” quando Z é utilizada para aproximar S . Quando as séries são normalizadas como histogramas, a desigualdade de Gibbs assegura que a distância KL será sempre não negativa (HAMMING, 1991). A distância de **Jeffrey** (Equação 4.18) é uma variante simétrica da KL. Outra variante simétrica da KL é a distância de **Jensen-Shannon** (Equação 4.19), cujo score é contido no intervalo $[0, \ln(2)]$ para distribuições de probabilidade (ou $[0, 1]$ quando se emprega $\log_2$ no seu cálculo). A raiz quadrada dessa distância, $\sqrt{d_{\text{JS}}}$, é uma métrica para distribuições de probabilidades.

$$d_{\text{KL}}(S, Z) = \sum_{t=1}^n s[t] \ln \left(\frac{s[t]}{z[t]} \right) \quad (4.17)$$

$$d_{\text{Jef}}(S, Z) = \sum_{t=1}^n (s[t] - z[t]) \ln \left(\frac{s[t]}{z[t]} \right) \quad (4.18)$$

$$d_{\text{JS}}(S, Z) = \frac{1}{2} \sum_{t=1}^n \left[s[t] \ln \left(\frac{2s[t]}{s[t] + z[t]} \right) + z[t] \ln \left(\frac{2z[t]}{s[t] + z[t]} \right) \right] \quad (4.19)$$

A distância de **Bhattacharyya** (Equação 4.20) é derivada do coeficiente de Bhattacharyya,

$$BC(S, Z) = \sum_{x \in S} \sqrt{P(S = x) \cdot P(Z = x)}.$$

O coeficiente de Bhattacharyya, também denominado coeficiente de fidelidade, é um indicador do grau de sobreposição (similaridade) entre duas distribuições de probabilidade. Algumas distâncias derivadas desse coeficiente compõem as distâncias da **família de Bhattacharyya**. A

distância de **Hellinger** (Equação 4.21) é uma medida derivada desse coeficiente e é uma métrica sobre um espaço de distribuição de probabilidades. A distância **squared-chord** (Equação 4.22) é uma distância bastante utilizada na paleontologia e a distância de **Matusita** (Equação 4.23) é uma transformação da distância de Bhattacharyya para o intervalo $[0, \sqrt{2}]$.

$$d_{\text{Batt}}(S, Z) = -\ln \left(\sum_{t=1}^n \sqrt{s[t]z[t]} \right) \quad (4.20)$$

$$d_{\text{Hel}}(S, Z) = 2\sqrt{1 - \sum_{t=1}^n \sqrt{s[t]z[t]}} \quad (4.21)$$

$$d_{\text{SqChrd}}(S, Z) = \sum_{t=1}^n \left(\sqrt{s[t]} - \sqrt{z[t]} \right)^2 \quad (4.22)$$

$$d_{\text{Mat}}(S, Z) = \sum_{t=1}^n \sqrt{2 - 2 \sum_{t=1}^n s[t]z[t]} \quad (4.23)$$

4.5.1 Tratamento especial de medidas não elásticas

Como mencionado na Seção 3.9.3, a z -normalização é o mecanismo de normalização mais comum em trabalhos de classificação de séries temporais por similaridade. Entretanto, algumas funções de distância exigem não negatividade das observações ou, uma restrição ainda mais forte, que a área sob a curva seja unitária. Por exemplo, a função χ^2 quadrada pode resultar em “distâncias” negativas se as séries possuírem observações negativas. Ilustra-se esse fato com o gráfico de espalhamento da Figura 29. Quando as séries são z -normalizadas, essa distância produz resultados péssimos. Ao se normalizar as séries para o intervalo $[0, 1]$, entretanto, essa distância produz resultados comparáveis à euclidiana.

Além disso, todas as distâncias baseadas em entropia de Shannon e no coeficiente BC são potencialmente inadequadas para séries cujas observações podem assumir valores negativos. A distância Kullback-Leibler, por exemplo, pode produzir termos que envolvem o logaritmo de um número negativo, produzindo um valor de “distância” que contém uma componente imaginária. Mais ainda, as distâncias baseadas no coeficiente BC (Bhattacharyya e afins) só garantem resultados válidos quando a área sob a curva de ambas as séries forem unitárias. Recomenda-se utilizar normalização ao intervalo $[0, 1]$ para todas as distâncias da família χ^2 e normalização como histogramas para todas as distâncias baseadas no coeficiente BC. Ambos os métodos de normalização podem ser adequados para as distâncias baseadas em entropia de Shannon, dependendo da aplicação. Nesta Tese, utilizou-se normalização $[0, 1]$ para a distância Kullback-Leibler e histograma para as demais distâncias dessa família.

Figura 29 – Acurácia do 1-NN utilizando distância χ^2 quadrada com e sem normalização no intervalo $[0, 1]$. No gráfico de espalhamento à *esquerda*, o eixo vertical refere-se ao classificador 1-NN com χ^2 quadrada quando as séries estão *z*-normalizadas; o eixo horizontal ao classificador 1-NN tradicional. À *direita*, as mesmas distâncias são utilizadas, mas as séries estão normalizadas no intervalo $[0, 1]$.

Finalmente, as distâncias χ^2 de Pearson, Neyman χ^2 , χ^2 quadrada e Kullback-Leibler são assimétricas. Elas são baseadas em coeficientes nos quais um objeto de interesse (S) é comparado contra um objeto de referência (Z). Nesses casos, deve-se tomar cuidado para que a série rotulada seja empregada como o objeto de referência.

É importante ressaltar que, mesmo quando as séries são normalizadas adequadamente, algumas funções podem envolver termos nos quais as relações não são definidas ou levam a um escore final que não está em $\mathbb{R}$. Por exemplo, o coeficiente χ^2 de Pearson pode originar termos que envolvem divisão por zero ou até mesmo do indefinições como $\frac{0}{0}$. Na Tabela 7 são apresentadas situações nas quais termos podem levar a problemas de implementação, bem como sugestões de resolução.

Tabela 7 – Termos indefinidos ou não numéricos nas funções de distâncias e possíveis soluções. Os índices das observações são omitidos por simplicidade; d é o valor sugerido para substituir o termo “problemático” e $x \neq 0$.

Problema	Afetando	Nos termos	Possível resolução
$0 \div 0$	Todas as funções cujos termos envolvem frações, incluindo $\ln\left(\frac{0}{0}\right)$.	$s = z = 0$	Assumir $d = 0$, pois as observações são idênticas.
$x \div 0$	χ^2 de Pearson	$s = 0$	Assumir $d = 0$, ignorando o termo, ou $d = \max(s, z)$.
	Neyman χ^2	$z = 0$	
$0 \ln 0$	Kullback-Leibler e Jeffrey	$z = 0$	Assumir $d = 0$, pois $\lim_{x \rightarrow 0} x \ln x = 0$.
	Jensen-Shannon	$s = 0$ ou $z = 0$	
$x \ln(0)$	Jeffrey	$s = 0$	Assumir $d = 0$, pois é uma variante simétrica da distância de Kullback-Leibler.
$x \ln\left(\frac{x}{0}\right)$		$z = 0$	

Note que essas sugestões de correção são tomadas com o objetivo de maximizar a utilidade das distâncias na comparação de séries temporais. Por exemplo, termos do tipo $x \div 0$ poderiam ser resolvidos atribuindo-se $d = \infty$, mas isso teria como consequência uma distância “infinita” entre duas séries. Em vez disso, pode-se simplesmente descartar esse caso ($d = 0$) ou

atribuir a ele o valor da maior observação, ignorando o termo nulo. A melhor decisão, nesse e em outros casos, depende de análise específica para cada aplicação.

4.6 Distâncias elásticas

Em contraste com as medidas *lockstep*, caracterizadas por um alinhamento síncrono no tempo entre as observações, as **distâncias elásticas** promovem um alinhamento que pode variar com os dados das séries. Distâncias elásticas são muitas vezes mais adequadas do que distâncias *lockstep*, pois podem oferecer invariâncias como invariância a oclusão e a *warping* local.

Entretanto, distâncias elásticas podem ser desvantajosas nos seguintes casos:

1. Quando, por qualquer motivo, necessita-se de uma distância de fácil implementação: distâncias elásticas envolvem conceitos mais complexos;
2. Quando os recursos computacionais são restritos: distâncias elásticas tendem a ser computacionalmente mais complexas. Embora haja métodos para mitigar ou eliminar esse problema em muitos domínios, podem existir situações extremas nas quais limitações de memória e/ou tempo computacional impedem o uso dessas distâncias;
3. Quando é difícil definir parâmetros: algumas distâncias elásticas utilizam parâmetros e, em muitos casos, pode ser preferível utilizar um método que não depende de parâmetros;
4. Quando a distância elástica promove invariância excessiva: em algumas circunstâncias, a distância elástica pode distorcer excessivamente as séries ou ignorar segmentos excessivos das séries. Isso é tipicamente mitigado ou solucionado associando-se parâmetros restritivos às funções de distância, como é o caso da DTW, mas nem sempre é desejável.

A distância elástica mais comum em aplicações envolvendo séries temporais é a *Dynamic Time Warping* e suas variantes. Esta e outras distâncias elásticas são apresentadas a seguir.

4.6.1 *Dynamic Time Warping*

Uma dificuldade na análise de sinais digitais é a ausência de uniformidade do eixo temporal. Esse fenômeno, denominado *warping* local (Seção 2.4, pág. 43), é especialmente comum em aplicações que envolvem reconhecimento de fala, pois a fala é naturalmente associada a mudanças de ritmo, pausas e interrupções. Para lidar com esse tipo de sinal, diferentes algoritmos de programação dinâmica foram propostos, nas décadas de 1970 e 1980, para promover alinhamento não linear entre sinais de áudio (SAKOE; CHIBA, 1978).

A **distância DTW** (*Dynamic Time Warping*) pode ser considerada uma “extensão” da distância euclidiana que suporta invariância a *warping* local. Sua versão mais simples, ocasionalmente denominada DTW “irrestrita”, obtém a distância entre duas séries temporais pelo *warping*

que minimiza a distância entre os sinais (BERNDT; CLIFFORD, 1994). O classificador 1-NN utilizando essa distância é usualmente denominado 1-NN DTW.

Informalmente, podemos considerar que o alinhamento DTW obtido “esticando” e “contraíndo” uma das séries no eixo do tempo, como mostrado para as duas séries na Figura 30. Considere que S e Z têm suas observações contidas no mesmo intervalo e a série S é mostrada “acima” de Z para destacar as projeções, representadas pelas sombras. Em A, uma única observação da série S é “esticada” e projetada sobre um segmento de Z . Em B, um segmento de S é “contraído” e projetado sobre uma única observação de Z .

Figura 30 – Exemplo de *warping* da série S sobre a série Z . Uma única observação de S , indicada pela seta A, é projetada sobre uma subsequência em Z . A subsequência de S , indicada pela seta B, é projetada em um único ponto de Z .

Um *warping* de S para Z é uma sequência de pares ordenados (i, j) que indica o alinhamento de uma observação $s[i]$ com uma observação $z[j]$. Para que um *warping* seja válido, as seguintes propriedades devem ser respeitadas:

- *Monotonicidade*: um alinhamento (i, j) deve ser necessariamente sucedido por um alinhamento (i', j') no qual $i' \geq i$ e $j' \geq j$;
- *Continuidade*: a sequência deve contemplar todas as observações de S e Z ;
- *Restrição de fronteira*: os alinhamentos $(1, 1)$ e (n, m) devem estar presentes na sequência.

O *warping* ideal entre duas séries é aquele que minimiza a função de custo $\gamma(n, m)$, definida pela seguinte relação:

$$\gamma(i, j) = \begin{cases} 0, & \text{se } i = j = 0 \\ \infty, & \text{se } i \neq j \text{ e } ij = 0 \\ (s[i] - z[j])^2 + \min \begin{pmatrix} \gamma(i-1, j), \\ \gamma(i, j-1), \\ \gamma(i-1, j-1) \end{pmatrix}, & \text{para os demais casos} \end{cases} \quad (4.24)$$

Com base nessa relação, a distância DTW entre uma série $S = (s[1], s[2], \dots, s[n])$ e uma série $Z = (z[1], z[2], \dots, z[m])$ é definida simplesmente como

$$d_{\text{DTW}}(S, Z) = \sqrt{\gamma(n, m)}. \quad (4.25)$$

Essa relação poderia ser transcrita, de maneira ingênua, como um algoritmo recursivo. Porém esse algoritmo produz uma explosão combinatorial e resulta em uma ordem de complexidade impraticável. Podemos definir um algoritmo $O(nm)$ para a distância DTW utilizando uma tabela de memoização para calcular, incrementalmente, os valores da relação de custo, como ilustrado na Figura 31. Para as séries S e Z , de tamanhos n e m , a matriz $M_{n+1 \times m+1}$ é inicialmente preenchida com os custos dados por $\gamma(1, 1)$, $\gamma(0, j)$ e $\gamma(i, 0)$. A partir daí, calculamos o custo das células adjacentes a $(1, 1)$ e preenchemos o restante da matriz de acordo com a relação γ . O “caminho” destacado na matriz contém os pares de observações que participam do *warping* ideal. Esse procedimento é descrito formalmente no Algoritmo 4.1.

Figura 31 – Exemplo de uso de memoização no cálculo da distância DTW. O caminho destacado na matriz à direita corresponde ao alinhamento ideal das séries S e Z à esquerda—adaptado de (SILVA, 2014).

O Algoritmo 4.1 possui complexidade de tempo e de espaço quadráticas com relação ao tamanho da maior série. Para conjuntos de dados com muitas sequências longas, tanto o tempo quanto o espaço necessário para calcular a distância DTW podem ser proibitivos.

Podemos facilmente reduzir a complexidade de espaço para $O(n)$ observando que, no laço interno que se inicia na linha 7 do Algoritmo 4.1, apenas as linhas $M(i, \cdot)$ e $M(i-1, \cdot)$ são consultadas. Desse modo, podemos modificar o algoritmo para guardar apenas a linha “anterior” no cálculo dos custos da linha “atual”. Ao final da iteração do laço interno, a linha “atual” passa a ser a linha “anterior”.

Quanto à complexidade temporal, não há algoritmo conhecido capaz de encontrar a distância DTW entre com complexidade melhor que $O(n^2)$ no pior caso. Entretanto, existem diversas estratégias para reduzir o tempo médio de uma aplicação que realiza múltiplos cálculos da distância DTW. Por exemplo, podemos empregar abandono antecipado no cálculo da distância entre uma série S e seu vizinho mais próximo. Observe que a função de custo é monotônica, de modo que o menor valor encontrado em uma linha $M(i, \cdot)$ da matriz não pode ser menor que o menor valor encontrado na linha $M(i+1, \cdot)$. No processo de busca por um vizinho mais próximo,

Algoritmo 4.1: Distância *Dynamic Time Warping*

1 **função** $d = \text{dtw}(S, Z)$ **Parâmetros de Entrada:**
 S : série temporal com n observações

 Z : série temporal com m observações

Parâmetros de Saída:
 d : distância DTW entre S e Z

2 $M \leftarrow$ matriz com $n + 1$ linhas e $m + 1$ colunas

3 $M(i, 0) \leftarrow \infty$ para todo $i = 1, 2, \dots, n$

4 $M(0, j) \leftarrow \infty$ para todo $j = 1, 2, \dots, m$

5 $M(0, 0) \leftarrow 0$

6 **para** i **de** 1 **até** n **faça**

7 **para** j **de** 1 **até** m **faça**

8 $c \leftarrow \min \begin{pmatrix} M(i-1, j), \\ M(i, j-1), \\ M(i-1, j-1) \end{pmatrix}$

9 $M(i, j) \leftarrow (s[i] - z[j])^2 + c$

10 **fim**

11 **fim**

12 $d \leftarrow \sqrt{M(n, m)}$

podemos sempre guardar um valor $(d_{\text{BSF}})^2$ que indica o quadrado da distância entre S e o melhor candidato encontrado nas buscas anteriores. Ao verificarmos se uma série candidata Z' é vizinho mais próximo de S , podemos abandonar a busca assim que verificarmos que o menor valor de custo na linha “atual” é superior a $(d_{\text{BSF}})^2$.

O Algoritmo 4.1 apresenta a denominada “versão irrestrita” da distância DTW. Modificações da DTW incluem associar uma janela de *warping* e atribuir pesos distintos na relação de recorrência.

4.6.2 Distância DTW com janela de *warping*

Um problema decorrente do uso da distância DTW é o “*warping* patológico”. Isso ocorre quando a função de *warping* produz distorção excessiva do eixo temporal ao minimizar a distância entre as séries, como mostrado na Figura 32. Uma possível solução para esse problema é empregar uma banda de *warping* (RATANAMAHATANA; KEOGH, 2004a). A banda de *warping* estabelece restrições sobre a relação de custo γ , impedindo alinhamentos entre pares de observações muito distantes no eixo do tempo.

A banda de *warping* mais simples é a **banda Sakoe-Chiba** (SAKOE; CHIBA, 1978). A banda Sakoe-Chiba é equivalente a uma janela deslizante de tamanho fixo δ . Durante o cálculo da distância, um alinhamento entre as observações $s[i]$ e $z[j]$ só é permitido se ambas podem ser posicionadas simultaneamente sob a janela. A relação γ pode ser facilmente adaptada para a relação γ_S com uma restrição adicional, como mostrado na Equação 4.26. Uma forma de

Figura 32 – Exemplo de *warping* patológico. A distância DTW distorce excessivamente o eixo temporal para minimizar a distância entre as séries (*esquerda*); uma restrição de *warping*, como a banda Sakoe-Chiba, reduz esse efeito (*direita*)—imagem obtida de (SILVA, 2014).

visualizar a banda Sakoe-Chiba é como uma delimitação na matriz de memoização em torno da diagonal secundária (Figura 33-*esquerda*): a área hachurada se estende δ linhas acima e abaixo da diagonal secundária e todos os alinhamentos fora dessa área possuem custo infinito.

$$\gamma_S(i, j) = \begin{cases} \infty, & \text{se } |i - j| > \delta \\ \gamma(i, j), & \text{caso contrário} \end{cases} \quad (4.26)$$

O **paralelogramo de Itakura** (ITAKURA, 1975) restringe a função de *warping* ao espaço delimitado por pares de retas paralelas com inclinação θ e $\frac{1}{\theta}$ (Figura 33-*direita*). O paralelogramo de Itakura é mais restritivo para as observações iniciais e finais das séries, equivalente ao emprego de janelas maiores no “meio” do *warping*. Para duas séries S e Z de tamanhos n e m , tais que $m \geq n$, a função de custo com paralelogramo de Itakura é obtida pela relação γ_I , observado-se a restrição $\theta \geq m/n$.

Figura 33 – Demonstração da banda de Sakoe-Chiba e do paralelogramo de Itakura—figura adaptada de (RATANAMAHATANA; KEOGH, 2004b).

$$\gamma_l(i, j) = \begin{cases} \infty, & \text{se } i > \lceil 1 + \theta(j-1) \rceil \\ \infty, & \text{se } i > \left\lceil \frac{j+\theta n-m}{\theta} \right\rceil \\ \infty, & \text{se } i < \lfloor \theta(j-m) + n \rfloor \\ \infty, & \text{se } i < \left\lfloor \frac{j+\theta-1}{\theta} \right\rfloor \\ \gamma(i, j) & \text{para os demais casos} \end{cases} \quad (4.27)$$

O Algoritmo 4.1 pode ser modificado para considerar apenas pares de observações que são válidos de acordo com as relações γ_S e γ_l . Por exemplo, podemos obter a banda Sakoe-Chiba alterando a condição do laço interno na linha 7 para “ **j de $\max(1, j-\delta)$ até $\min(j+\delta, n)$** ”. No caso extremo em que $\delta = 0$, obtemos a distância euclidiana (contanto que $n = m$). Similarmente, a distância euclidiana é um caso particular do paralelogramo de Itakura para $\theta = 1$. A complexidade de tempo da distância DTW é amortizada pela banda Sakoe-Chiba e pelo paralelogramo de Itakura: no primeiro caso a complexidade é $O(n\delta)$ e, no segundo, $O(n\theta)$, considerando-se que n é o tamanho da série mais longa.

A desvantagem decorrente do uso de uma banda de *warping* é a introdução de um parâmetro. Em geral, algoritmos desprovidos de parâmetros são preferíveis, a introdução de parâmetros é acompanhada de viés (KEOGH; LONARDI; RATANAMAHATANA, 2004). Não obstante, a necessidade de se prevenir *warping* patológico tipicamente justifica o emprego de alguma forma de restrição. Diversos autores têm relatado que o classificador 1-NN tem desempenho notório, na classificação de séries temporais, quando associado à distância DTW com uma banda Sakoe-Chiba (ASSENT *et al.*, 2009; KEOGH; RATANAMAHATANA, 2005; XI *et al.*, 2006). O tamanho ideal da janela de *warping* é normalmente encontrado com validação cruzada *leave-one-out* ou 10-fold CV sobre os dados de treinamento.

4.6.3 Distância DTW com pesos

Outra forma de se restringir o *warping* é associar pesos aos alinhamentos dependendo dos índices das observações. A *Weighted Dynamic Time Warping*, DTW com pesos ou simplesmente WDTW é uma variante da DTW que utiliza uma função de peso proporcional à diferença de fase entre as observações que fazem parte de um alinhamento (JEONG; JEONG; OMITAOMU, 2011), favorecendo alinhamentos entre observações mais próximas no eixo do tempo. A relação de custo da WDTW pode ser descrita como na Equação 4.28, na qual $f_\gamma(\varphi)$ é uma função de

peso a ser definida sobre a diferença de fase $\varphi = |i - j|$.

$$\gamma_W(i, j) = \begin{cases} 0, & \text{se } i = j = 0 \\ \infty, & \text{se } i \neq j \text{ e } ij = 0 \\ f_\gamma(|i - j|) \cdot (s[i] - z[j])^2 + \min \begin{pmatrix} \gamma_W(i - 1, j), \\ \gamma_W(i, j - 1), \\ \gamma_W(i - 1, j - 1) \end{pmatrix}, & \text{para os demais casos} \end{cases} \quad (4.28)$$

Uma simples função de peso poderia ser $f_\gamma(\varphi) = 1 + \varphi$. Para compreender como a função de peso afeta o cálculo da distância, considere o alinhamento DTW das séries mostradas na Figura 34. Na parte superior da figura é mostrada a série $S = (0; 0,5; 0,5)$ e, na inferior, $Z = (0; 0,2; 0,4)$. A distância euclidiana entre essas séries é $d_{\text{Euc}} = 0,32$. A distância DTW entre elas é $d_{\text{DTW}} = 0,24$.

Figura 34 – Exemplo de alinhamento DTW entre duas séries.

A distância DTW entre as duas séries é menor que a distância euclidiana porque a soma das distâncias entre as observações envolvidas nos alinhamentos (1,2) e (2,3) é menor que a distância entre as observações do alinhamento (2,2). A capacidade de “evitar” o alinhamento de alto custo (2,2) e encontrar o caminho mais favorável $\langle (1,1), (1,2), (2,3), (3,3) \rangle$ é a característica da função de *warping*.

Quando empregamos uma função de peso, associamos um custo adicional aos alinhamentos de maior distorção. No caso da função de peso $f_\gamma(\varphi) = 1 + \varphi$, alinhamentos adjacentes são associados a um custo $f_\gamma(1) = 2$. Para o exemplo da Figura 34, isso significa que os alinhamentos (1,2) e (2,3) só serão selecionados como parte do *warping* de S para Z se o *dobro* da soma das distâncias das observações envolvidas neles for menor que a distância entre as observações do alinhamento (2,2). Como se pode observar na Tabela 8, isso não ocorre para as séries do exemplo.

O objetivo desse exemplo é mostrar que a função de peso deve ser cautelosamente definida para evitar o *warping* patológico sem tornar muito elevado o custo do *warping* nos casos em que ele é desejado. Jeong, Jeong e Omitaomu (2011) consideram uma função de peso logística envolvendo três parâmetros: o valor máximo da função de peso (w_{max}), um fator de “centralidade” m_c e um fator de linearidade $g \geq 0$. Quando $g = 0$, todos os alinhamentos recebem peso $f_\gamma = 0,5 \cdot w_{\text{max}}$ e conforme $g \rightarrow \infty$, a função logística se aproxima de uma função

Tabela 8 – Distâncias entre pares das observações das séries mostradas na Figura 34 (*esquerda*) e distâncias corrigidas com uma função de peso (*direita*). As células hachuradas são aquelas que fazem parte do alinhamento ideal em cada caso. A distância final do alinhamento à *esquerda* é $\sqrt{0,06} = 0,24$ e à *direita* é $\sqrt{0,10} = 0,32$.

	$f_\gamma = 1$			$f_\gamma = 1 + i-j $		
$s[3]$	0,25	0,09	0,01	0,75	0,18	0,01
$s[2]$	0,25	0,09	0,01	0,50	0,09	0,02
$s[1]$	0	0,04	0,16	0	0,08	0,48
	$z[1]$	$z[2]$	$z[3]$	$z[1]$	$z[2]$	$z[3]$

degrau para a qual $f_\gamma = 1$ se a distância do alinhamento no eixo temporal é $|i-j| \geq m_c$. As curvas na Figura 35 mostram como o peso varia, em função razão do valor máximo $w_{\max}$, para diferentes valores de g . A relação que produz a função logística mostrada na Figura 35 é dada pela Equação 4.29.

$$f_\gamma(\varphi) = \frac{w_{\max}}{1 + e^{-g(\varphi - m_c)}} \quad (4.29)$$

Note que, em (JEONG; JEONG; OMITAOMU, 2011), a relação de recorrência da distância DTW é considerada sobre um espaço de Minkowski e, portanto, os autores acrescentam um parâmetro p . Os autores também empregam a função de peso no mesmo espaço de Minkowski, de modo que a relação se torna (simplicadamente)

$$\gamma_W(i, j) = (f_\gamma(\varphi)(s[i] - z[j]))^p \cdot \min(\cdot).$$

Entretanto, os autores não apresentam argumentos ou evidências de que um valor $p \neq 2$ produza resultados melhores de classificação. Experimentos realizados durante a execução deste projeto sugerem que não há razão para justificar o acréscimo desse parâmetro. Também não se verifica vantagem em utilizar os parâmetros $w_{\max}$ e m_c . De fato, Jeong, Jeong e Omitaomu (2011) argumentaram que o parâmetro $w_{\max}$ não afetou os resultados. Os efeitos do parâmetro m_c (“centralidade”) podem ser ajustados com a variação do parâmetro g (linearidade).

Todavia, os parâmetros existentes na Equação 4.29 precisam ser definidos. Nesta Tese, foi considerado $w_{\max} = 1$. O fator de “centralidade” foi definido com base no seguinte raciocínio: para séries de tamanhos n e m , o menor alinhamento ocorre entre quaisquer pares de observações (i, i) e o maior alinhamento ocorre entre $(1, \max(n, m))$. Dessa forma, o fator de “centralidade” foi definido como $m_c = \frac{n+m}{4}$, tomando como base para isso o “maior alinhamento médio”, $(1, \frac{n+m}{2})$. A função de peso logística simplificada se torna a seguinte.

$$f_\gamma(\varphi) = \frac{1}{1 + e^{-g(\varphi - \frac{n+m}{4})}} \quad (4.30)$$

Figura 35 – Curvas para a função de peso logística com diferentes valores do fator de linearidade g . As curvas foram produzidas para os parâmetros $w_{\max} = 1$ e $m_c = 10$.

4.6.4 Longest Common Subsequence

Sejam C_1 e C_2 duas cadeias de símbolos tomados de um domínio discreto. O problema da subsequência comum mais longa (LCS, do inglês *Longest Common Subsequence*) consiste em encontrar a mais longa cadeia C_c que seja simultaneamente uma subsequência de C_1 e C_2 . Trata-se de um problema clássico de computação que encontra aplicações em diversas áreas. Por exemplo,

- em sistemas de controle de versão, a diferença entre dois programas P_1 e P_2 é o conjunto de todas as linhas que não pertencem à subsequência comum mais longa entre eles; e
- na biologia molecular, sequências genéticas são representadas como cadeias de caracteres sobre o alfabeto $\Sigma = \{A, T, C, G\}$. A subsequência mais longa comum a duas cadeias genéticas reflete o grau de similaridade dessas duas cadeias.

É importante destacar que, neste contexto, uma subsequência C_c de C_1 não é necessariamente uma cadeia de símbolos contíguos em C_1 . Por exemplo, considere as sequências genéticas $G_1 = \text{ACTG}$ e $G_2 = \text{CGAA}$. A subsequência mais longa comum entre elas é a cadeia CG. Essa

cadeia ocorre como uma *substring* em $\underline{\text{CGAA}}$ e como uma subsequência em $\underline{\text{ACTG}}$. Além disso, essa é a cadeia mais longa de símbolos sobre o alfabeto genético que é simultaneamente uma subsequência de G_1 e G_2 .

O conceito de subsequência comum mais longa em séries temporais foi explorado inicialmente por Das, Gunopulos e Mannila (1997) e uma variante normalizada foi apresentada por Vlachos, Kollios e Gunopulos (2002). A **distância LCSS** é uma função de distância que promove um alinhamento elástico entre duas séries temporais S e Z , admitindo que algumas observações não façam parte do alinhamento. O melhor alinhamento desse tipo entre duas séries é a subsequência comum mais longa entre elas.

Como as observações das séries são contínuas, a subsequência comum mais longa entre S e Z é encontrada com base um limiar ε que estabelece um casamento entre duas observações $s[i]$ e $z[j]$ se $|s[i] - z[j]| \leq \varepsilon$. Além disso, o alinhamento restringe o *warping* máximo que pode ser empregado sobre as séries, de modo que um casamento entre $s[i]$ e $z[j]$ só é possível se $|i - j| \leq \delta$. Sob essas restrições, dadas duas séries temporais $S = (s[1], s[2], \dots, s[n])$ e $Z = (z[1], z[2], \dots, z[m])$, a distância LCSS entre duas séries é definida como na Equação 4.31, na qual $\text{LCSS}_{\varepsilon, \delta}$ é o tamanho da subsequência mais longa entre S e Z .

$$d_{\text{LCSS}}(S, Z) = 1 - \frac{\text{LCSS}_{\varepsilon, \delta}(S, Z)}{\min(n, m)} \quad (4.31)$$

Para encontrar $\text{LCSS}_{\varepsilon, \delta}$ comum mais longa, considere o operador $\text{Head}(S)$ que retorna a “cabeça” da série temporal S ,

$$\text{Head}(S) = s[1 : n-1], \quad (4.32)$$

que é a subsequência de todas as observações de S , exceto a última. Note que, para uma série que contenha uma única observação, $\text{Head}(S) = ()$ é uma “subsequência vazia”.

Com base nessas definições, o tamanho da subsequência comum mais longa a duas séries temporais pode ser definido pela seguinte relação de recorrência:

$$\text{LCSS}_{\varepsilon, \delta}(S, Z) = \begin{cases} 0, & \text{se } S = () \text{ ou } Z = () \\ 1 + \text{LCSS}_{\varepsilon, \delta}(\text{Head}(S), \text{Head}(Z)), & \text{se } |s[n] - z[m]| \leq \varepsilon \text{ e } |n - m| \leq \delta \\ \max \left(\begin{array}{l} \text{LCSS}_{\varepsilon, \delta}(\text{Head}(S), Z), \\ \text{LCSS}_{\varepsilon, \delta}(S, \text{Head}(Z)) \end{array} \right), & \text{para os demais casos} \end{cases} \quad (4.33)$$

Um algoritmo trivial para encontrar a subsequência comum mais longa entre duas séries poderia ser a transcrição da relação 4.33 como um algoritmo recursivo. Porém, esse algoritmo

possui complexidade de tempo exponencial. Podemos empregar uma matriz de memoização semelhante àquela que utilizamos para calcular a distância DTW e obter um algoritmo de complexidade temporal $O(n\delta)$ —no qual n é o tamanho da série mais longa—, mostrado no Algoritmo 4.2.

Algoritmo 4.2: Distância LCSS

1 **função** $d = \text{lcss}(S, Z, \varepsilon, \delta)$

Parâmetros de Entrada:

S : série temporal com n observações

Z : série temporal com m observações

ε : limiar de distância máxima do casamento entre $s[i]$ e $z[j]$

δ : limite de *warping* permitido para o casamento entre $s[i]$ e $z[j]$

Parâmetros de Saída:

d : distância LCSS entre S e Z

2 $M \leftarrow$ matriz com $n + 1$ linhas e $m + 1$ colunas

3 Preencha todas as células de M com $M(i, j) = 0$

4 **para** i **de** 1 **até** n **faça**

5 **para** j **de** $\max(1, i - \delta)$ **até** $\min(m, i + \delta)$ **faça**

6 **se** $|s[i] - z[j]| \leq \varepsilon$ **então**

7 $M(i, j) \leftarrow M(i - 1, j - 1) + 1$

8 **senão**

9 $M(i, j) \leftarrow \max(M(i - 1, j), M(i, j - 1))$

10 **fim**

11 **fim**

12 **fim**

13 $d \leftarrow 1 - \frac{M(n, m)}{\min(n, m)}$

4.6.5 Distância de edição

A distância de edição entre duas sequências A e B é o número mínimo de operações necessárias para modificar a sequência $A = (a_1, a_2, \dots, a_n)$ e transformá-la na sequência $B = (b_1, b_2, \dots, b_m)$. Não existe uma definição única de distância de edição, mas normalmente considera-se que são válidas as operações propostas por Levenshtein (1966), que consistem em inserção, remoção e substituição.

Por exemplo, dada a sequência $A = \text{EULER}$, podemos obter a sequência $B = \text{ESCHER}$ através das substituições $U \rightarrow S$ e $L \rightarrow C$ e da adição do símbolo H . Esse é o número mínimo de operações de Levenshtein necessárias para realizar essa transformação, portanto a distância de edição entre essas sequências é 3.

A distância de Levenshtein foi proposta por Levenshtein (1966) como o número máximo de erros de inserção, remoção e substituição que um algoritmo seria capaz de detectar em uma mensagem codificada em símbolos binários. A extensão dessa distância para um alfabeto discreto é trivial, como evidenciado pelo exemplo anterior. Para séries temporais, Chen, Özsu e Oria

(2005) propuseram utilizar um limiar de distância ε , obtendo a distância EDR (*Edit Distance on Real Sequences*, ou distância de edição em sequências reais). Considere o custo do casamento entre as observações $s[i]$ e $z[j]$,

$$c_{\text{match}}(i, j) = \mathbb{I}[|s[i] - z[j]| > \varepsilon], \quad (4.34)$$

no qual $\mathbb{I}[\cdot]$ é a função indicadora (pág. 3.6). A distância EDR pode ser obtida pela relação de recorrência EDR_ε , na qual Head é o mesmo operador “cabeça” utilizado na Equação 4.33.

$$\text{EDR}_\varepsilon(S, Z) = \begin{cases} |Z|, & \text{se } S = () \\ |S|, & \text{se } Z = () \\ \min \begin{pmatrix} \text{EDR}_\varepsilon(\text{Head}(S), \text{Head}(Z)) + c_{\text{match}}(n, m), \\ \text{EDR}_\varepsilon(\text{Head}(S), Z) + 1, \\ \text{EDR}_\varepsilon(S, \text{Head}(Z)) + 1 \end{pmatrix}, & \text{para os demais casos} \end{cases} \quad (4.35)$$

Podemos empregar uma estratégia de programação dinâmica e utilizar uma matriz de memoização para encontrar a distância de edição entre duas séries temporais com complexidade de tempo $O(nm)$. Tal estratégia é apresentada no Algoritmo 4.3.

Note que a relação de recorrência da distância de edição é bastante similar à relação de recorrência empregada na distância LCSS. De fato, a distância LCSS é um tipo de distância de edição na qual a operação de substituição não é permitida. Note que a matriz de memoização do Algoritmo 4.3 é iniciada com a primeira linha e a primeira coluna representando os dois primeiros casos da relação de recorrência $\text{LCSS}_{\varepsilon, \delta}$. As células $M(0, j)$ e $M(0, i)$ representam a distância de edição entre a sequência vazia e alguma sequência qualquer de tamanho i ou j , que é o custo de inserir (ou, no sentido inverso, remover) todos os elementos da sequência não vazia.

Assim como a distância DTW, a distância EDR também permite o uso de técnicas de abandono rápido. O Algoritmo 4.3 percorre a matriz de memoização da primeira até à última linha. Como nenhuma operação é capaz de reduzir o custo de edição, então nenhuma sequência de operações encontrada na i -ésima iteração pode ter custo inferior às operações da iteração $i - 1$. Portanto, podemos guardar um valor d_{BSF} que indica a menor distância entre uma série de interesse e o vizinho mais próximo “atual”. Se o menor valor encontrado em uma linha for $d_i \geq d_{\text{BSF}}$, então podemos abandonar o cálculo da distância.

4.7 Avaliação experimental

Dentre os principais tópicos de pesquisa desta Tese, o emprego de diferentes funções de distâncias com o classificador 1-NN é provavelmente o mais explorado na literatura. Muitos desses estudos concentram-se em comparar o desempenho de distâncias elásticas.

Algoritmo 4.3: Distância de edição (EDR)

```

1 função  $d = \text{edr}(S, Z, \varepsilon)$ 
  Parâmetros de Entrada:
    S: série temporal com  $n$  observações
    Z: série temporal com  $m$  observações
     $\varepsilon$ : limiar de distância máxima do casamento entre  $s[i]$  e  $z[j]$ 
  Parâmetros de Saída:
     $d$ : distância EDR entre  $S$  e  $Z$ 
2  $M \leftarrow$  matriz com  $n + 1$  linhas e  $m + 1$  colunas
3 para  $i$  de 1 até  $n$  faça
4    $M(i, 0) \leftarrow i$ 
5 fim
6 para  $j$  de 1 até  $m$  faça
7    $M(0, j) \leftarrow j$ 
8 fim
9  $M(0, 0) \leftarrow 0$ 
10 para  $i$  de 1 até  $n$  faça
11   para  $j$  de 1 até  $m$  faça
12      $\text{CustoSubstituição} \leftarrow M(i - 1, j - 1) + c_{\text{match}}(i, j)$ 
13      $M(i, j) \leftarrow \min \begin{pmatrix} \text{CustoSubstituição}, \\ M(i - 1, j) + 1, \\ M(i, j - 1) + 1 \end{pmatrix}$ 
14   fim
15 fim
16  $d \leftarrow M(n, m)$ 

```

A seguir, apresenta-se uma avaliação das funções de distância elásticas selecionadas para a realização deste trabalho. Além disso, foram repetidos experimentos envolvendo as distâncias não elásticas discutidas nas seções Seção 4.4 e Seção 4.5. Alguns resultados apresentados nesta Tese diferem daqueles que foram anteriormente relatados em (GIUSTI; BATISTA, 2013), devido à adoção de novos métodos de normalização e tratamento de casos especiais, como discutido na Seção 4.5.1.

4.7.1 Ajuste de parâmetros

Todas as funções elásticas exploradas neste trabalho possuem parâmetros que podem ser ajustados para cada caso de aplicação. Para os experimentos conduzidos neste trabalho, os parâmetros foram ajustados, utilizando validação cruzada *leave-one-out* sobre os exemplos de treinamento do particionamento canônico do repositório UCR. O espaço de busca considerado para cada função de distância é apresentado na Tabela 9.

O parâmetro δ , para a distância LCSS, especifica a separação máxima entre duas observações na subsequência comum mais longa. Para definir o espaço de busca, inicialmente verificou-se, através de experimentos preliminares, que valores muito elevados de δ são típica-

Tabela 9 – Espaço de busca de parâmetros relacionados às distâncias elásticas. Na primeira linha, n é o número de observações da série.

Distância	Parâmetro	Descrição	Espaço de busca
DTW	δ	Largura da janela Sake-Chiba	0%, 1%, 2%, ..., 100% de n
WDTW	g	Linearidade da função logística	0,1; 0,2 ... 0,9
LCSS	ε	Limiar de casamento	0,01; 0,02; 0,05; 0,1; 0,2; 0,5
LCSS	δ	Separação máxima para casamento	*
EDR	ε	Limiar de casamento	0,01; 0,02 ... 0,1; 0,2; 0,3 ... 0,6

mente ineficazes. Portanto, foi estabelecido como valor máximo do espaço de busca $\delta_{\max} = 0,3n$. Note que um casamento entre observações separadas por $\delta_{\max}$ observações significa que pelo menos 30% das observações de *ambas* as séries não estará presente na subsequência comum mais longa. Portanto o espaço de busca foi definido como a união dos conjuntos $\{1, 2 \dots 10\}$ e $\{5\%, 10\%, 15\% \dots 30\%\}$ do tamanho da série.

Os melhores parâmetros foram utilizados para avaliar as distâncias nos conjuntos de teste. Os valores de acurácia de teste e os parâmetros selecionados para todos os conjuntos de dados são apresentados no apêndice (Seção A.3, pág. 221).

4.7.2 Utilizando diferentes funções de distância

Uma importante consequência das observações feitas anteriormente é que existem diferentes funções de distância adequadas para diferentes domínios de aplicação. Durante a execução deste projeto, foi realizado um estudo envolvendo 48 funções de distância, incluindo versões invariantes a complexidade das distâncias euclidiana e DTW (GIUSTI; BATISTA, 2013). Como mencionado no Capítulo 2, existem diversas estratégias para estimar a complexidade de séries temporais (PRATI; BATISTA, 2012). No estudo apresentado em (GIUSTI; BATISTA, 2013), foi empregada a abordagem original de (BATISTA; WANG; KEOGH, 2011), na qual a complexidade de uma série temporal é obtida “alongando-se” a série até que ela esteja “achatada”. A intuição por trás dessa medida é que séries complexas possuem muitos “vales” e “picos”, de modo que a soma dos comprimentos dos seus segmentos correlaciona positivamente com a complexidade. Séries mais complexas, quando “achatadas”, produzem sequências mais longas, como ilustrado na Figura 12 (pág. 46).

O estimador de complexidade de uma série temporal S com n observações é dado pela Equação 4.36.

$$CE(S) = \sqrt{\sum_{t=1}^{n-1} (s[t] - s[t+1])^2} \quad (4.36)$$

A utilização desse estimador se dá como explicado na Seção 2.4: um fator de correção de complexidade é obtido pela razão da complexidade entre duas séries. Quanto maior fora

diferença de complexidade entre as séries, maior será o fator de correção de complexidade, “penalizando” as séries com uma distância mais elevada.

Os resultados obtidos nesse experimento, para o domínio temporal, são sumarizados na Tabela 10. Para esse experimento, foram utilizados os 45 conjuntos de dados originais do repositório UCR que estavam disponíveis à época da publicação do artigo. Note que, nesse experimento, foram empregadas algumas funções de distância além das que são apresentadas neste capítulo. A razão dessa divergência se deve ao fato de que algumas funções de distância foram abandonadas em etapas posteriores da execução deste trabalho devido, principalmente, à baixa acurácia verificada nesse experimento. Uma descrição completa de todas as funções de distância pode ser encontrada em (CHA, 2007).

Tabela 10 – Sumário: média e desvio padrão de acurácia de distâncias.

Distância	Média	Desv. padrão	Distância	Média	Desv. padrão
DTW + CID	0,8184	0,1374	Max Symmetric χ	0,2720	0,2145
DTW	0,8103	0,1419	Pearson	0,2698	0,2033
DTW (sem janela)	0,7905	0,1519	Jensen Difference	0,2665	0,2503
Euclidiana + CID	0,7774	0,1460	Topsøe	0,2664	0,2503
Minkowski L_p	0,7660	0,1417	Jeffrey	0,2659	0,2480
Lorentzian	0,7640	0,1484	Square Chord	0,2647	0,2516
Manhattan	0,7627	0,1501	Wavehedges	0,2580	0,2119
Média L_1/L_∞	0,7619	0,1500	Tanimoto	0,2561	0,2440
Dice	0,7529	0,1479	K Divergence	0,2503	0,2084
Jaccard	0,7529	0,1479	Vicis Wave Hedges	0,2501	0,2262
Euclidiana	0,7507	0,1493	χ^2 Quadrada	0,2417	0,2255
Chebyshev (L_∞)	0,6719	0,1892	Additive Symm χ^2	0,2409	0,2184
Hellinger	0,6408	0,1932	Canberra	0,2408	0,2093
Divergência	0,5128	0,1989	Neyman	0,2394	0,2264
Soergel	0,4963	0,1843	Kullback-Leibler	0,2394	0,2080
Clark	0,2938	0,2296	Taneja	0,2390	0,2150
Kulczynski	0,2905	0,2304	Emanon 3	0,2377	0,2298
Motyka	0,2778	0,2461	Min Symmetric χ	0,2371	0,2213
Sørensen	0,2778	0,2461			

Os resultados apresentados na Tabela 10 tinham como objetivo apenas elencar um conjunto de distâncias adequadas, no caso médio, para uma análise posterior. Assim, foram selecionadas algumas distâncias para a realização de testes de hipótese. O teste de hipótese de Friedman foi realizado com $\alpha = 0,05$ considerando 8 medidas em 42 conjuntos de dados. A hipótese nula de que todas as funções de distância produzem classificadores idênticos foi rejeitada. Em seguida, foi conduzido um teste post-hoc de Nemenyi. O resultado desse teste é apresentado como um gráfico de distância crítica na Figura 36.

O teste de hipótese indica que a distância DTW com invariância a complexidade produz classificadores substancialmente melhores que todas as outras distâncias, exceto a distância euclidiana com invariância a complexidade. Uma conclusão desse experimento é que a distância DTW é tipicamente superior, mas distâncias não elásticas de menor complexidade computacional podem ser competitivas.

Figura 36 – Gráfico de diferença crítica para as funções de distância no domínio temporal.

4.7.2.1 “Atualização” dos experimentos

Ao longo da execução deste projeto, o repositório UCR foi expandido com novos conjuntos de dados e novas funções de distância elásticas foram introduzidas aos experimentos. Por esse motivo, os experimentos anteriormente apresentados foram revistos, com algumas mudanças:

- Não foram reavaliadas as variantes CID, pois os experimentos anteriores já haviam sugerido que a invariância a complexidade é capaz de melhorar o desempenho do classificador 1-NN;
- Foram acrescentadas as distâncias elásticas EDR, LCSS e WDTW, utilizando o procedimento de ajustes de parâmetros relatado na Seção 4.7.1;
- Foi selecionado um subconjunto relevante das distâncias cujos resultados foram apresentados na Tabela 10. As distâncias foram escolhidas de modo a promover um conjunto de classificadores diversos e foram descartadas funções que produziram classificadores muito pobres no experimento anterior;
- Foram adotadas as práticas de normalização e tratamento de casos especiais, apresentadas na Seção 4.5.1, para as funções de distância Pearson, χ^2 Quadrada, Jeffrey, Kullback-Leibler, Hellinger e Bhattacharyya.

Tabela 11 – Sumário: média e desvio padrão de acurácia de distâncias (segundo experimento).

Distância	Média	Desv. padrão	Distância	Média	Desv. padrão
EDR	0,7695	0,1620	Dice	0,7120	0,1754
LCSS	0,7637	0,1600	Cosseno	0,7120	0,1754
DTW	0,7630	0,1606	Jaccard	0,7120	0,1754
WDTW	0,7604	0,1694	Kullback-Leibler	0,7055	0,1737
Lorentzian	0,7241	0,1692	Hellinger	0,7015	0,1701
Manhattan	0,7223	0,1712	Canberra	0,6979	0,1676
Média L_1/L_∞	0,7219	0,1710	Pearson	0,6968	0,1812
Sørensen	0,7201	0,1698	Bhattacharyya	0,6939	0,1860
χ^2 Quadrada	0,7165	0,1711	Jeffrey	0,6654	0,2140
Euclidiana	0,7120	0,1754	Chebyshev (L_∞)	0,6546	0,1944

Uma primeira observação que pode ser feita é que, nos resultados apresentados na Tabela 10, o classificador 1-NN obteve o mesmo desempenho com as distâncias euclidiana, Dice, cosseno e Jaccard. Nos experimentos anteriores, a distância cosseno não havia sido empregada, e o valor médio da distância euclidiana foi ligeiramente inferior ao das outras duas. Essa discrepância se deve ao fato de que os resultados apresentados na Tabela 10 são baseados no particionamento canônico do repositório UCR antes de sua recente atualização. À época, alguns conjuntos de dados não estavam normalizados. Com a normalização de todos os conjuntos do repositório UCR, as quatro medidas produzem resultados idênticos.

Foi realizado um teste de hipótese de Friedman com as distâncias que apresentaram resultados médios superiores aos da distância euclidiana. A estatística-F verificada foi 17,58 e o valor crítico para $\alpha = 0,95$ é com 9 e 756 graus de liberdade é 1,89. O teste de Friedman rejeitou a hipótese nula de que os algoritmos são idênticos e, portanto, foi conduzido um teste post-hoc de Nemenyi para verificar quais algoritmos podem ser considerados distintos com significância estatisticamente significativa. O resultado desse teste é apresentado como um gráfico CD na Figura 37.

Figura 37 – Gráfico de distância crítica para distâncias elásticas e não elásticas.

4.7.2.2 O “1-NN dinâmico”

Diante da variabilidade de desempenho mostrada pelas diferentes funções de distância, a questão “lógica” é se é possível selecionar uma distância ideal para cada conjunto de dados. Uma forma de se realizar isso é utilizar validação cruzada nos exemplos de treinamento e escolher a melhor distância. Esse classificador que, para fins de discussão podemos denominar “1-NN dinâmico”, certamente terá desempenho superior se as distâncias forem consistentes entre si no treino e no teste.

Um experimento foi realizado para verificar essa hipótese e os resultados são apresentados na Figura 38. O 1-NN dinâmico seleciona a melhor distância não elástica dentre as apresentadas na Tabela 10. O 1-NN elástico considera todas as distâncias da tabela simultaneamente.

No apêndice deste trabalho, apresenta-se uma tabela contendo os valores de acurácia obtidos por esses dois classificadores, bem como as funções de distância selecionadas (Seção A.4, pág. 223). A distância selecionada com maior frequência para o 1-NN dinâmico foi a distância média entre as normas L_1 e L_∞ (17 conjuntos), seguida da χ^2 Quadrada (11) e da Canberra (10). A distância euclidiana foi escolhida em 9 conjuntos. Quando o classificador 1-NN dinâmico

Figura 38 – Desempenho do classificador 1-NN com seleção de distâncias. À esquerda, o desempenho do classificador 1-NN selecionando a melhor dentre as distâncias não elásticas consideradas neste trabalho. À direita, o desempenho do 1-NN selecionando a melhor dentre todas as distâncias. Note que os classificadores de referência são distintos nos dois casos.

pôde considerar também as distâncias elásticas nos experimentos, elas foram escolhidas para 78 dos 85 conjuntos de dados. A distância Chebyshev (norma L_∞), uma função de distância relativamente pouco utilizada e que apresentou o pior desempenho médio no experimento anterior, foi selecionada apenas duas vezes pelo 1-NN elástico. Em ambos os casos (conjuntos *Adiac* e *ProximalPhalanxOutlineCorrect*) a acurácia dessa distância nos exemplos de treinamento superou tanto o classificador 1-NN quanto o 1-NN DTW. Já a distância euclidiana foi utilizada pelo 1-NN elástico apenas na classificação do conjunto *NonInvasiveFetalECG2*. Essa é apenas uma observação que reforça a importância de se considerar distâncias e representações tipicamente vistas como não convencionais.

As distâncias Dice, Jaccard, Hellinger e Cosseno não foram escolhidas para nenhum conjunto de dados nesse experimento. Essas distâncias serão novamente exploradas no próximo capítulo, associadas a outras representações.

4.8 Considerações finais

O 1-NN é um classificador baseado em instâncias que depende fortemente de uma função de distância capaz de representar vizinhanças compatíveis com o conceito que se deseja aprender. O emprego de diversidade de distâncias na classificação de séries temporais é um problema bem conhecido e estudado. De fato, pode-se argumentar que o uso de funções de distância como método para definir espaços alternativos de classificação tem sido mais estudado do que o uso de representações de séries.

Por ora, as funções de distância estão sendo avaliadas apenas no domínio temporal, no qual medidas elásticas produzem resultados definitivamente superiores, no caso médio, com relação às distâncias não elásticas. A distância DTW, especialmente quando associada a uma

janela Sakoe-Chiba, é normalmente considerada o estado-da-arte.

No próximo capítulo, será introduzido o conceito de mudança de representação. Na literatura, é comum que seja utilizada a distância euclidiana nesse domínio. Neste trabalho, serão combinados os conceitos de mudança de representação com diversidade de distâncias. Levanta-se a hipótese de que, para algumas representações, existem domínios nos quais o uso de distâncias diferentes da euclidiana pode levar a resultados superiores, em termos de acurácia de classificação.

TRANSFORMAÇÃO DE SÉRIES

5.1 Considerações iniciais

AS OBSERVAÇÕES DE UMA SÉRIE TEMPORAL são, tipicamente, um registro de um fenômeno mensurável que varia com o tempo. Alternativamente, elas podem representar um fenômeno adjacente, como os valores de distância verificados enquanto se percorre um contorno durante um processo de serialização (como definido na Seção 2.2). O espaço no qual essas séries estão representadas é conhecido como **domínio temporal**.

Uma série temporal pode ser transformada do domínio temporal para um domínio de representação alternativos, nos quais seus valores representam conceitos que não são diretamente observáveis no domínio temporal. A pertinência desse fato para esta Tese é a existência de representações que explicitam atributos discriminativos que não são claros no domínio original.

Provavelmente o caso mais ilustrativo é o de classificação de sinais de áudio, como dígitos falados por meio de um telefone. No domínio temporal, as séries representam a magnitude de uma onda mecânica (som) ao longo do tempo. Nessa representação, a distinção entre sinais que contêm dígitos distintos envolve interação complexa entre as observações das séries. Isto é, os atributos de interesse que permitem identificar a classe associada às séries estão “profundamente embutidos” na inter-relação entre suas observações. Nesse caso, pode-se transformar a série para um domínio de frequências, no qual as observações descrevem o fenômeno por meio de uma grandeza muito mais próxima aos conceitos associados à fala. Nesse domínio, é possível extrair atributos mais significativos das séries ou mesmo classificá-las diretamente por meio do classificador 1-NN de forma mais eficaz do que no domínio temporal.

Esse tipo de transformação envolve uma substituição da relação temporal entre as observações por um outro tipo de relação de ordem. No entanto, a série transformada ainda apresenta as características fundamentais de uma série temporal, como definido na Seção 2.2 desta Tese. Isto é, as “observações” da série transformada para um domínio de frequências

representam uma relação de ordem entre componentes de frequência ordenados.

Nesta Tese, a utilização de diferentes domínios de representação foi um ponto fundamental de pesquisa. Foi verificado que, individualmente, nenhuma representação permite classificar de forma ótima séries temporais em todo tipo de aplicação. Entretanto, utilizar representações alternativas oferece diferentes “visões” que permitem explorar diferentes relações de vizinhança no espaço de decisão. Isso é particularmente eficiente para a construção de *ensembles* de classificadores, como será mostrado adiante neste texto.

Além disso, foi explorada uma estratégia para utilizar diversidade de representação na extração de características genéricas para classificação dissociada de um modelo específico. Os atributos de distância, baseados no conceito de “DTW-Features”, são apresentados na Seção 5.7 como uma estratégia promissora para construção de classificadores robustos.

Neste capítulo, são apresentadas algumas das representações estudadas. As primeiras representações são baseadas em frequência e correlação. Outras são baseadas em sumarização das observações no domínio temporal ou em abordagens de “dicionário”. Como a maioria das representações estudadas possuem parâmetros que devem ser ajustados para cada caso, na Seção 5.8.1 são apresentados resultados de experimentos realizados a fim de verificar as configurações adequadas para os conjuntos do repositório UCR. Na Seção 5.8.2 é apresentado um estudo no qual se verificou a utilidade de cada representação individualmente em comparação com as demais e em comparação com o classificador 1-NN no domínio temporal. Na Seção 5.8.3 são apresentados resultados experimentais que sugerem a eficácia dos atributos de distância na classificação de séries temporais. Finalmente, considerações finais são apresentadas na Seção 5.9.

5.2 Representações baseadas em frequências

Uma sinal pode ser decomposto em um espectro de frequências através de uma transformada de Fourier. O espectro de frequências é uma representação do sinal original em componentes periódicas de frequências distintas. Diversas aplicações podem ser conduzidas sobre o espectro de frequências de um sinal (BACHMAN; NARICI; BECKENSTEIN, 2000) e alguns sinais são singulares no sentido de que seus espectros apresentam características marcantes associadas a uma ou mais de suas componentes.

Existem diferentes tipos de transformada de Fourier: a transformada contínua de Fourier é utilizada quando o sinal ocorre continuamente ao longo do tempo e produz um espectro contínuo de frequências. A transformada de tempo discreto de Fourier (*discrete time Fourier transform*, DTFT) é empregada quando o sinal é discreto e produz um espectro contínuo de frequências. Finalmente, a **transformada discreta de Fourier** (*discrete Fourier transform*, DFT) é empregada em sinais discretos e produz um espectro discreto de frequências.

No escopo desta Tese, as séries são amostradas digitalmente e, portanto, correspondem a sinais discretos. O propósito de se utilizar espectro de frequências é obter domínios alternativos de classificação. Deseja-se que estes sejam igualmente discretos e aplicáveis nas mesmas situações em que se pode utilizar os sinais originais no domínio temporal. Portanto, neste trabalho utiliza-se a transformada discreta de Fourier, como definido a seguir.

Definição 5.1: Dada uma sequência $S = (s[1], s[2], \dots, s[n])$, $S \in \mathbb{C}^n$, sua transformada discreta de Fourier é definida pelos coeficientes $(f[0], f[1], \dots, f[n-1])$, dados pela relação

$$\hat{f}[k] := \sum_{t=0}^{n-1} x[t+1] \cdot e^{-2\pi i \frac{tk}{n}}, \quad (5.1)$$

na qual i é a unidade imaginária $i = \sqrt{-1}$.

Os coeficientes da transformada de Fourier são tipicamente denominados *coeficientes de Fourier*. Note que, para simplificar a relação 5.1, os coeficientes são indexados de $\hat{f}[0]$ a $\hat{f}[n-1]$. Para uma série de cardinalidade n , pode-se obter n coeficientes de Fourier. O coeficiente $\hat{f}[0]$ é denominado DC e corresponde à frequência nula. O coeficiente $\hat{f}[\lfloor \frac{n}{2} \rfloor]$ corresponde à frequência máxima que pode ser capturada pela transformada de Fourier, denominada frequência de Nyquist (LEIS, 2011). Como será mostrado na Seção 5.2.1, os demais coeficientes correspondem a frequências intermediárias.

Nesta Tese, são utilizadas duas representações baseadas em frequências e uma representação baseada em tempo-frequência (Seção 5.2.4). A literatura é vasta em livros e artigos que tratam dessas representações, portanto nesta Tese não exploraremos em profundidade todo o corpo teórico associado. Utilizaremos exemplos para explicar os conceitos importantes no emprego dessas representações na classificação de séries temporais. Na Figura 39, à direita, são mostradas 100 observações de duas séries temporais simples produzidas pela Equação 5.2:

$$x[t] := A \cdot \sin\left(2\pi f \frac{t-1}{f_s}\right) \quad (5.2)$$

A Equação 5.2 descreve o comportamento de um sinal com amplitude A e frequência f Hz, amostrado nos instantes $t = 1, 2, \dots, n$ a uma taxa de f_s observações por segundo.

As séries mostradas na Figura 39-direita foram produzidas pela Equação 5.2 com uma taxa de amostragem $f_s = 100$ Hz. As séries possuem 100 observações, portanto foram amostradas durante um segundo nos instantes $t_1 = 0/100s, t_2 = 1/100s, \dots, t_{100} = 99/100s$. A componente de menor frequência possui $f = 4$ Hz e $A = 1$; a de maior, $f = 6$ Hz e $A = 0,7$. A sobreposição (soma) dessas séries produz o sinal mostrado na Figura 39-esquerda. Como se trata apenas de um exemplo numérico sem correspondência a qualquer fenômeno observado ou simulado, os valores das observações são escalares.

Figura 39 – Série temporal artificial com duas componentes senoidais.

5.2.1 Representação DFT

A representação mais simples que se deriva da transformada de Fourier é obtida quando os próprios coeficientes de Fourier são utilizados para representar a série. Nesse caso, a série temporal é representada em um espaço de decisão que extrapola a definição apresentada na Seção 2.2, pois as observações potencialmente contêm componentes imaginárias não nulas. Nesta Tese, essa representação é denominada **representação DFT**.

Definição 5.2: Dada uma série temporal $S = (s[1], s[2], \dots, s[n])$ e sua transformada de Fourier $\hat{F}$ (Definição 5.1), sua representação DFT é

$$DFT(S) = (f[1], f[2], \dots, f[n]), \quad (5.3)$$

no qual $f[k] = \hat{f}[k - 1]$.

Uma vez que a DFT é uma transformada de tempo-discreto, os coeficientes de Fourier não dizem respeito a valores específicos de frequências, mas a intervalos de componentes de frequências denominados *bins*. Para uma série temporal com n observações amostrada a uma taxa de amostragem f_s Hz, cada *bin* corresponde a uma faixa de $\frac{f_s}{n}$ Hz. Tipicamente, associa-se cada *bin* ao valor de frequência central ou a um valor marginal desse intervalo. Nesse caso, os *bins* de $f[1]$ até aproximadamente $f[n/2]$ podem ser associados às frequências centrais $(k - 1) \cdot \frac{f_s}{n}$. Como mencionado na seção anterior, a maior frequência que pode ser representada por essa transformação está no *bin* “central”, cujo índice é $t_n = \lfloor \frac{n}{2} \rfloor + 1$. As demais observações, excetuando-se a componente DC ($f[1]$) são simétricas em torno dessa observação.

Podemos verificar isso com a série de exemplo da Figura 39-direita. Essa série foi obtida pela composição de duas senoides, com taxa de amostragem 100 Hz. A representação DFT dessa

série pode ser descrita completamente pela relação seguinte:

$$f[k] = \begin{cases} 12,43 + 48,43i & \text{se } k = 5, & (4 \text{ Hz}) \\ 12,88 + 32,54i & \text{se } k = 7, & (6 \text{ Hz}), \\ 12,88 - 32,54i & \text{se } k = 95, & (6 \text{ Hz}), \\ 12,43 - 48,43i & \text{se } k = 97, & (4 \text{ Hz}), \\ 0 & \text{para outros valores de } k. \end{cases}$$

Uma propriedade particularmente relevante da transformada de Fourier é sua ortogonalidade. A transformada de Fourier preserva a distância euclidiana:

$$d_{\text{euc}}(S, Z) = d_{\text{euc}}(\text{DFT}(S), \text{DFT}(Z)).$$

Além disso, verifica-se, para esse exemplo, que a maior parte da energia da série de exemplo concentrada nas componentes de menor frequência. Quando isso ocorre, um sinal pode ser aproximadamente caracterizado por um pequeno número de coeficientes de Fourier, permitindo *lower bounding* da distância euclidiana no domínio temporal ([AGRAWAL; FALOUTSOS; SWAMI, 1993](#)). Isso é empregado, por exemplo, em aplicações nas quais a representação DFT de uma série temporal é truncada, promovendo redução de dimensionalidade e permitindo indexação eficiente de séries temporais ([FALOUTSOS; RANGANATHAN; MANOLOPOULOS, 1994](#)).

Por outro lado, associar a representação DFT com a distância euclidiana não traz nenhum benefício para classificação de séries temporais em domínios alternativos, pois preserva os conceitos de vizinhança da distância euclidiana no domínio temporal. Pode-se empregar outras funções de distância na representação DFT para as quais a transformada não é ortogonal. Entretanto, como os coeficientes de Fourier podem não estar contidos em $\mathbb{R}$, muitas funções de distância selecionadas neste trabalho precisariam ser adaptadas. Assim, a representação DFT é utilizada neste trabalho apenas como passo intermediário para obtenção da representação espectral.

5.2.2 Representação espectral

As componentes obtidas pela transformada de Fourier são caracterizadas por frequência, amplitude e fase. A frequência e a amplitude dizem respeito ao período e energia de cada componente. A fase diz respeito à “posição” da componente no instante inicial e sua interpretação mais simples consiste em considerar o quanto ela está desviada de uma senoide $s[t] = \sin(t)$. Na Figura 40 são mostradas três componentes que possuem mesma frequência e amplitude, mas fases distintas—uma representação ideal das fases presentes no sistema elétrico brasileiro.

Existem aplicações nas quais a informação de fase pode ser de grande importância. Em outros casos, apenas as frequências são relevantes. Por exemplo, no problema de classificação de curvas de luzes (Seção 2.5), a diferença de fase é geralmente considerada uma distorção indesejada do processo de amostragem. Nesse caso pode-se transportar as séries para um

Figura 40 – Representação ideal das fases presentes no sistema elétrico brasileiro. Cada fase possui frequência de 60 Hz e está defasada em um ângulo de 120° ou -120° com respeito às outras duas. A amplitude representa a tensão em Volts e diferentes configurações de tensão aparecem em diferentes estágios do transporte de energia ao longo do sistema.

domínio de representação no qual as observações derivam apenas da informação de amplitude e são invariantes à fase.

Na formulação mostrada na Definição 5.1, os coeficientes de Fourier são valores complexos que assumem a forma geral $f[k] = a + bi$. Dado um coeficiente, sua amplitude pode ser obtida pela Equação 5.4. Esse valor de amplitude não está necessariamente associado a uma unidade de medida. Normalmente, a intensidade de um sinal é mensurada em decibéis e é possível normalizar os valores de $A[k]$ para essa escala. Entretanto, para classificação por similaridade, essa não é uma etapa necessária, bastando que todas as séries sejam representadas na mesma escala—ainda que desconhecida.

$$A[k] = |f[k]| = \sqrt{a^2 + b^2} \quad (5.4)$$

O conjunto ordenado das amplitudes de um sinal pode ser denominado espectro de amplitudes ou espectro de frequências. Nesta Tese, essa representação será denominada **representação espectral** ou **representação FS** (*frequency spectrum*).

Definição 5.3: Dada uma série temporal $S = (s[1], s[2], \dots, s[n])$, $S \in \mathbb{C}^n$, sua representação espectral é

$$FS(S) = (A[1], A[2], \dots, A[n]), \quad (5.5)$$

no qual a sequência $(A[1], A[2], \dots, A[n])$ é obtida a partir das amplitudes de cada componente espectral de acordo com a Equação 5.4.

Na Figura 41 mostra-se a representação FS da série de exemplo apresentada na Seção 5.2. As componentes são aproximadamente nulas em quase toda a extensão da série transformada,

exceto nos picos associados às observações $A[5]$, $A[7]$, $A[95]$ e $A[97]$. Estas correspondem às componentes de frequências 4 Hz e 6 Hz.

Figura 41 – Representação FS da série mostrada na Figura 39-esquerda.

A representação espectral pode ser particularmente eficaz em domínios de aplicação que exigem invariância a fase. Ela também pode ser utilizada para discriminar classes que são caracterizadas pela amplitude de certas componentes de frequência ou para identificar possíveis padrões sazonais em uma série temporal. Para ilustrar, considere a série temporal na Figura 42-esquerda. Os dados referentes a essa série consistem de mensurações do nível médio mensal de precipitação de chuva na cidade de Lavras, em Minas Gerais (PARMEZAN; BATISTA, 2014). A série possui 384 observações referentes a um período de 32 anos entre janeiro de 1966 e dezembro de 1997. Na Figura 42-direita, mostra-se apenas a “primeira metade” da sua representação espectral.

Figura 42 – Exemplo de utilização da representação FS para expor sazonalidade. A série à esquerda foi obtida pela anotação da média mensal do nível de precipitação pluvial na cidade de Lavras (MG). Sua representação FS à direita mostra uma forte componente periódica de 12 observações.

Não existe critério estrito para definir quais componentes de frequências devem ser consideradas “distintivas”. No caso da Figura 42-direita, observa-se dois pontos que superam todos os demais em uma ordem de magnitude. O primeiro, que corresponde à frequência

0, reflete o valor médio das observações no domínio temporal e não indica sazonalidade. O coeficiente $A[32]$, por outro lado, sugere uma componente de frequência dominante na série original. Considerando-se a taxa de amostragem (mensal) e o número de amostras (384 meses), esse coeficiente sugere que a série Lavras é caracterizada por uma forte componente periódica com duração de aproximadamente 12,5 meses¹—uma conclusão bastante próxima do que seria esperado.

5.2.3 Short-time Fourier transform

As representações FS e DFT são adequadas para análise das frequências presentes em um sinal, mas não permitem identificar os momentos em que essas frequências ocorrem. Por exemplo, considere a série na Figura 43-esquerda. As observações dessa série foram obtidas pela Equação 5.2 em procedimento semelhante à série mostrada na Figura 39: as primeiras 100 observações correspondem a um segundo de um sinal com frequência $f = 5$ Hz; as demais corresponde a um segundo de um sinal com $f = 30$ Hz. Seu espectro de frequências, como mostrado na Figura 39-direita, apresenta dois picos associados a essas componentes. Note a presença de componentes espúrias. É possível reduzir o número de componentes espúrias aumentando-se o número de observações utilizadas na transformada (por exemplo, adicionando observações nulas ao sinal). Entretanto, o espectro de frequências é incapaz de identificar que as componentes ocorrem em instantes distintos no tempo e nenhum tipo de pré-processamento é capaz de contornar isso.

Figura 43 – Ilustração da ausência de informação de tempo no espectro de frequências. Na parte *esquerda* da figura, uma série que contém uma componente de 4 Hz nos instantes $t \in [0, 99]$ e uma componente de 6 Hz nos instantes $t \in [100, 199]$. No lado *direito*, seu espectro de frequências (desconsiderando a parte simétrica redundante).

Para se estudar a evolução das componentes de frequência ao longo do tempo, surgem as técnicas de análise de tempo-frequência. Uma dessas técnicas é a transformada “curta” de Fourier (*short-time Fourier Transform, STFT*). A STFT é obtida com o auxílio de uma janela (JURADO;

¹ Um sinal S com $|S| = n$ é decomposto em componentes de $n + 1$ frequências distintas. Seja Δt o intervalo entre duas amostras no domínio temporal, a representação f_s de um sinal corretamente amostrado tem resolução $1/n\Delta t$. Para todo $k \in [1, N + 1]$, a k -ésima componente possui frequência $k-1/n\Delta t$ e período $n\Delta t/k-1$ (CONDON; RANSOM, 2016).

SAENZ, 2002), produzindo uma representação dos *bins* em diferentes instantes. A evolução das frequências ao longo do tempo é normalmente representada como um *espectrograma*, como mostrado na Figura 44. No eixo horizontal, o espectrograma indica valores de frequência associados aos *bins*. No eixo vertical, a cor ou luminosidade do espectrograma indica a energia associada aos *bins*.

Na Figura 44, são mostrados quatro espectrogramas da série composta pelas subsequências de 5 e 30 Hz, mostrada na Figura 43-esquerda. Cada espectrograma foi obtido através da transformada “curta” de Fourier com um tamanho de janela distinto. De cima para baixo e, da esquerda para a direita, os espectrogramas mostrados foram produzidos com janelas de tamanhos $w = 10$, $w = 20$, $w = 40$ e $w = 80$ observações.

Figura 44 – Espectrogramas obtidos com transformada “curta” de Fourier.

Acompanhando os eixos verticais dos espectrogramas, podemos estudar a evolução temporal das componentes de frequência na série. Como esperado, observa-se que a série pode ser dividida em dois “estágios”. A transição entre esses estágios ocorre aproximadamente em $t = 1s$, quando a série deixa de ser dominada por uma componente de baixa frequência e passa a ser dominada por uma componente de alta frequência.

O objetivo da Figura 43 é ilustrar o compromisso na precisão com que a representação tempo-frequência pode ser localizada no tempo e na frequência (BACHMAN; NARICI; BECKENSTEIN, 2000). Em todos os espectrogramas da figura, é possível observar que existe uma mudança de comportamento da série por volta de $t = 1s$, a partir do qual a série deixa de ser caracterizada por componentes de baixa frequência e passa a ser caracterizada por componentes de alta frequência. Existe um período de “transição” no qual todas as frequências aparentam

ser aproximadamente igualmente distribuídas. Quando a janela é curta ($w = 10$), essa faixa de “transição” é breve e a mudança de comportamento da série é bem clara no eixo temporal, mas existe uma incerteza considerável com relação aos valores de frequência dominantes nas séries. Quando a janela é longa ($w = 80$), as faixas mais escuras indicam claramente a existência de duas componentes, uma de frequência aproximadamente $f = 5$ Hz e outra de frequência aproximadamente $f = 30$ Hz. Por outro lado, essas componentes são menos localizadas no domínio temporal e todo o intervalo desde $t = 0,8s$ a $t = 1,2s$ apresenta informação que contradiz o conhecimento prévio de como a série foi construída. Os outros dois espectrogramas mostram compromissos intermediários de resolução de tempo contra resolução de frequência.

Devido a esse compromisso, a transformada “curta” de Fourier é bastante sensível ao tamanho da janela. Em aplicações nas quais a resolução no tempo é mais importante, utiliza-se janelas mais curtas. Em aplicações nas quais a resolução na frequência é mais importante, utiliza-se janelas mais longas. Entretanto, em qualquer caso não é possível representar todas as componentes de frequência de forma apropriada com uma única janela.

5.2.4 Transformada discreta de wavelets

A transformada de *wavelets* é uma abordagem de **análise multirresolução** ou MRA (*multi resolution analysis*) para contornar os problemas associados ao uso de uma janela fixa na obtenção da representação tempo-frequência. A transformada de *wavelets* emprega uma janela dinâmica que permite estabelecer diferentes graus de compromisso na resolução dos coeficientes. Uma janela curta é utilizada para extrair as componentes de alta frequência, favorecendo a resolução temporal. Essa janela é expandida para a extração das componentes de baixa frequência, favorecendo a resolução no domínio de frequências. Embora isso não elimine a incerteza associada à representação do sinal, a transformada de *wavelets* favorece a informação mais útil para cada componente.

As janelas são caracterizadas por uma família de funções derivadas de uma *wavelet* mãe e uma função de dilatação ou escala (BACHMAN; NARICI; BECKENSTEIN, 2000). A transformada de *wavelets* é obtida através da convolução do sinal com essas funções. O exemplo mais simples de *wavelet* mãe é a *wavelet* de Haar:

$$\Psi(t) = \begin{cases} 1 & \text{se } t \in [0, \frac{1}{2}), \\ -1 & \text{se } t \in [\frac{1}{2}, 1), \\ 0 & \text{para demais valores de } t. \end{cases} \quad (5.6)$$

Para a qual a função de dilatação é simplesmente

$$\phi(t) = \begin{cases} 1 & \text{se } t \in [0, 1), \\ 0 & \text{caso contrário.} \end{cases} \quad (5.7)$$

A função de dilatação ou escala é utilizada para produzir os diferentes níveis da *wavelet* mãe que são utilizadas na transformação em coeficientes de tempo-frequência. No caso da *wavelet* de Haar, a composição das funções $\Phi(t)$ e $\phi(t)$ produz a função geradora

$$\Psi_{j,k}(t) = \sqrt{2^j} \cdot \Psi(2^j t - k), \quad (5.8)$$

na qual j é um parâmetro de escala e k é um parâmetro de translação. O conjunto dessas funções define uma base. Em particular, a *wavelet* de Haar define uma base ortonormal, o que a torna apropriada para tarefas como indexação de sinais e compressão de som e imagem (POPIVANOV; MILLER, 2002; CHAN; FU, 1999). Essa base pode ser utilizada para descrever uma série temporal,

$$f = \sum_{j,k \in \mathbb{Z}} \langle f, \Psi_{j,k} \rangle \Psi_{j,k}, \quad (5.9)$$

na qual o operador $\langle f, g \rangle$ representa o produto interno entre as funções f e g . Implicitamente, a Equação 5.9 promove uma convolução do sinal original com as funções derivadas da *wavelet* mãe, o que define a transformada contínua de *wavelets*. A operação de convolução equivale a percorrer a série com uma janela, cuja largura varia em função do coeficiente de escala ou dilatação j .

No caso particular em que o sinal e os coeficientes desejados são discretos, a **transformada discreta de wavelets** (DWT ou *discrete wavelet transform*) pode ser utilizada. A transformada discreta é semelhante à transformada contínua no sentido de que é obtida pela convolução do sinal original com um conjunto de funções base. Entretanto, a DWT pode ser obtida através de um processo iterativo envolvendo a aplicação de filtros passa-baixa e passa-alta (MALLAT, 1989).

Um filtro passa-baixa pode ser descrito como um operador que produz uma correlação do sinal original com um conjunto de baixas frequências, efetivamente eliminando as componentes de alta frequência do sinal de saída. Equivalentemente, um filtro passa-alta remove componentes de baixa frequência. Na DWT, os filtros são vetores em $\mathbb{R}^N$ cujos coeficientes derivam da função de dilatação relacionada à *wavelet*. Os filtros de *decomposição* são utilizados para decompor uma série em coeficientes tempo-frequência. Filtros de *composição* promovem a transformação inversa.

No caso da *wavelet* de Haar, os filtros passa-baixa e alta são, respectivamente, $h_{LP} = \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)$ e $h_{HP} = \left(\frac{-1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)$. Para uma série amostrada em f_s Hz, quando o sinal é convolvido com o filtro passa-baixa, obtém-se coeficientes referentes às componentes de frequências no intervalo $[0, \frac{f_s}{2}]$, denominadas **coeficientes de aproximação**. Quando o sinal é convolvido com o filtro passa-alta, o resultado refere-se às componentes de frequência no intervalo $[\frac{f_s}{2}, f_s]$, denominadas **coeficientes de detalhe**. O conjunto de coeficientes obtido através da convolução com cada filtro é um vetor de tamanho $|S| + |h_{LP}| - 1$. Isso significa o sinal resultante possui mais observações que o sinal original, mas apenas metade das componentes de frequência, de modo

que ele pode ser sub-amostrado. Isso é feito, por exemplo, descartando-se todos os coeficientes com índice ímpar ou par.

Através desse procedimento, obtém-se um **nível** da transformada discreta de *wavelets*. Os coeficientes de aproximação podem então ser submetidos aos mesmos filtros, produzindo níveis subsequentes da transformada. A cada passagem pelo filtro, a redução das frequências representadas pelos coeficientes implica aumento na resolução de frequências. Por outro lado, a sub-amostragem dos coeficientes resultantes equivale à dilatação da *wavelet* mãe e implica uma redução da resolução temporal. Essa configuração é denominada *filter bank*. O número máximo de níveis em um *filter bank* depende do número de observações da série original e da *wavelet* mãe utilizada. Para a *wavelet* de Haar, podemos obter até $\lfloor \log_2(n) \rfloor$ níveis.

Definição 5.4: *Seja S uma série temporal com n observações. Sejam h_{LP} e h_{HP} filtros associados a uma *wavelet*. Seja S_{D_1} o vetor que corresponde aos coeficientes de detalhe, obtidos pela convolução de S com o filtro h_{HP} e em seguida sub-amostrados. De maneira equivalente, o vetor S_{A_1} corresponde aos coeficientes de aproximação, obtidos por meio do filtro h_{LP} . Então a sequência $[S_{A_1}, S_{D_1}]$ corresponde ao primeiro nível da transformada discreta de *wavelets* (DWT) de S .*

A convolução do vetor S_{A_1} com os filtros produz, por sua vez, a sequência $[S_{A_2}, S_{D_2}]$, de modo que a sequência $[S_{A_2}, S_{D_2}, S_{D_1}]$ corresponde aos dois primeiros níveis da transformada. O detalhamento iterativo dos coeficientes de aproximação produz a DWT de S em k níveis: $[S_{A_k}, S_{D_k}, S_{D_{k-1}}, \dots, S_{D_1}]$.

Existem diversas famílias de *wavelets*, como as de Daubechies, as *coiflets* e as *symlets*. As principais características que distinguem essas famílias são suporte e o número de momentos nulos (UZINSKI, 2013). Quanto mais compacto for o suporte da *wavelet*, maior a resolução temporal da transformada e, quanto maior o número de momentos nulos, maior a probabilidade de reconstrução do sinal transformado.

Dentre aquelas que podem ser empregadas na transformada discreta, optou-se por utilizar neste trabalho as de Daubechies. As *wavelets* de Daubechies são uma família de *wavelets* que minimizam o número de coeficientes necessários para um certo número de momentos nulos. A *wavelet* de Haar é um caso particular de Daubechies com apenas um momento nulo. As Daubechies são tipicamente identificadas pelos termos Db1 (Haar), Db2, Db3 etc., representando o número de momentos nulos (ou pelos termos D2, D4, D6 etc., representando o número de coeficientes). A escolha de um filtro particular dessa família depende dos objetivos de sua aplicação, considerando-se os efeitos do número de coeficientes na resolução temporal e na regularidade do sinal obtido. Como não se assume qualquer aplicação específica neste trabalho, o número de momentos nulos foi selecionado através de validação cruzada nos exemplos de treinamento.

Finalmente, podemos definir a **representação em *wavelets*** ou a **representação DWT**:

Definição 5.5: Seja S uma série temporal com n observações e um parâmetro N que determina uma wavelet de Daubechies na forma DbN . Seja $|D|$ o número de coeficientes do filtro associado a DbN . O número máximo de iterações que podem ser empregadas em S para obter sua transformada discreta de wavelets é

$$k = \left\lfloor \log_2 \left(\frac{n}{|D| - 1} \right) \right\rfloor. \quad (5.10)$$

Assim, a representação DWT de S com parâmetro N é a sequência ordenada dos vetores $[S_{A_k}, S_{D_k}, S_{D_{k-1}}, \dots, S_{D_1}]$ obtidos pelo processo iterativo realizado de acordo com a Definição 5.4.

5.3 Representação PAA (Piecewise Aggregate Approximation)

Segundo (FU, 2011), uma forma natural, porém tipicamente ineficaz de se reduzir dimensionalidade de séries temporais consiste em sub-amostrar uma nova série a partir das observações de uma série já amostrada. Por exemplo, pode-se descartar todas as observações nas posições ímpares da série e reduzi-la à metade de sua cardinalidade original. Uma extensão natural desse procedimento seria segmentar a série e amostrar um ponto médio de cada segmento obtido (YI; FALOUTSOS, 2000). Essa representação de série temporal é também conhecida como *Piecewise Aggregate Approximation* ou PAA (KEOGH *et al.*, 2001). A **representação PAA** é uma importante representação em séries temporais que promove redução de dimensionalidade com suporte a *lower bounding*.

Dada uma série temporal S de cardinalidade n , intenciona-se encontrar sua representação PAA de cardinalidade $N \ll n$. O primeiro passo consiste em segmentar a série com base nas subsequências $s[\alpha_1 : \omega_1]$, $s[\alpha_2 : \omega_2]$, $\dots$, $s[\alpha_N : \omega_N]$. Os coeficientes α_k e ω_k estabelecem os extremos dos segmentos e podem ser encontrados através das equações 5.11 e 5.12.

$$\alpha_k := \begin{cases} 1, & \text{se } k = 1 \\ (k-1) \frac{n}{N}, & \text{caso contrário} \end{cases} \quad (5.11)$$

$$\omega_i := k \frac{n}{N} \quad (5.12)$$

Por ora, vamos assumir que N é um divisor de n . Essa restrição garante que todos os segmentos são subsequências válidas de S e é assumida apenas para simplificar a discussão que se segue. Posteriormente nesta seção, a transformação será generalizada para situações em que o número de observações da série não é um múltiplo do número de segmentos.

Definição 5.6: Seja $S = (s[1], s[2], \dots, s[n])$ uma série temporal de tamanho n . Seja $N \in [1, n]$ um parâmetro que indica o número de segmentos desejados. Assumindo-se, sem perda de

generalidade, que N é um divisor de n , a representação PAA de S é a sequência

$$PAA(S) = (p[1], p[2], \dots, p[N]), \quad (5.13)$$

na qual cada coeficiente $p[k]$ equivale ao valor médio das observações da subsequência $s[\alpha_k : \omega_k]$, i.e.:

$$p[k] = \overline{s[\alpha_k : \omega_k]} := \frac{N}{n} \sum_{j=\alpha_k}^{\omega_k} s[j]. \quad (5.14)$$

Um exemplo dessa representação é dado na Figura 45. A série temporal original é representada pela curva ao fundo. As linhas verticais denotam os limites das subsequências resultantes da segmentação da série. A representação PAA da série consiste em oito observações, indicadas pelos “degraus” marcados com traço espesso na figura.

Figura 45 – Representação de uma série no domínio PAA—adaptado de (LIN *et al.*, 2007).

A série mostrada na Figura 45 possui, convenientemente, 128 observações. Isso significa que cada coeficiente PAA é calculado da média de oito observações. Na prática, entretanto, essa situação é excepcional. Quando N não é um múltiplo de n , cada coeficiente PAA deve ser calculado a partir da média de uma fração de n/N observações. Isso significa que algumas observações devem contribuir parcialmente no cálculo dos coeficientes vizinhos, como mostrado na Figura 46.

A primeira observação da série está totalmente contida no intervalo do primeiro coeficiente, mas a segunda observação possui interseção com o limite entre o primeiro e o segundo coeficientes. Mais especificamente, a observação $s[2]$ deve contribuir com peso $\frac{1}{3}$ ao primeiro coeficiente com peso $\frac{2}{3}$ ao segundo. Para integrar esses pesos à Equação 5.14, utiliza-se um abuso de notação no qual um intervalo pode começar ou terminar em um índice na forma $(a + \frac{a'}{N})$, no qual a e a' são inteiros, mas $\frac{a'}{N}$ não. A “observação” $s[a + \frac{a'}{N}]$ assume um dos seguintes valores:

$$s\left[a + \frac{a'}{N}\right] = \begin{cases} 1 - \frac{a'}{N}s[a+1], & \text{no começo da subsequência} \\ \frac{a'}{N}s[a+1], & \text{no fim da subsequência} \end{cases} \quad (5.15)$$

Como exemplo, demonstra-se os passos para encontrar os três primeiros coeficientes PAA quando uma série de 10 observações é segmentada em seis subsequências:

Figura 46 – Indexação de segmentos fracionados na transformação PAA. Na parte *superior* da figura, uma série com dez observações é segmentada em cinco subsequências; todos os segmentos começam e terminam em índices inteiros. Na parte *inferior*, uma série com mesma cardinalidade é segmentada em três subsequências; é necessário considerar “índices” fracionados—adaptado de (LIN *et al.*, 2007).

Primeiro coeficiente $p[k = 1]$: de acordo com as Equações 5.11 e 5.12, encontra-se $\alpha_1 = 1$ e $\omega_1 = \frac{10}{6}$. Reescrevendo ω_1 , verifica-se que o primeiro coeficiente deve ser calculado sobre as “observações” $s[1]$ e $s[1 + \frac{4}{6}]$. Uma vez que a segunda “observação” está no fim da subsequência, tem-se, de acordo com a relação 5.15, que

$$s\left[1 + \frac{4}{6}\right] = \frac{4}{6}s[1 + 1] = \frac{4}{6}s[2].$$

Dessa forma, o primeiro coeficiente é

$$p[1] = \frac{6}{10} \left(s[1] + \frac{4}{6}s[2] \right);$$

Segundo coeficiente $p[k = 2]$: encontra-se $\alpha_2 = \frac{10}{6}$ e $\omega_2 = \frac{20}{6}$. Reescrevendo esses valores encontra-se $\alpha_2 = 1 + \frac{4}{6}$ e $\omega_2 = 3 + \frac{2}{6}$. Assim, de acordo com a relação 5.15, o segundo coeficiente é

$$p[2] = \frac{6}{10} \left(\frac{2}{6}s[2] + s[3] + \frac{2}{6}s[4] \right);$$

Terceiro coeficiente $p[k = 3]$: encontra-se $\alpha_3 = 3 + \frac{2}{6}$ e $\omega_3 = 5$. Note que ω_3 não indica uma observação de fato, portanto para este elemento da subsequência será utilizada a própria observação $s[\omega_3]$ em vez de consultar-se a relação 5.15. Assim, tem-se o terceiro coeficiente:

$$p[3] = \frac{6}{10} \left(\frac{4}{6}s[4] + s[5] \right).$$

O quarto e os demais coeficientes são calculados com base na subsequência $s[5 : 10]$. Nesse exemplo, os pesos envolvidos nos cálculos remanescentes serão idênticos aos mostrados nos três primeiros passos, alterando-se apenas os índices das observações envolvidas.

A principal finalidade da representação PAA é redução de dimensionalidade, sendo eficiente para índice de séries temporais. [Keogh et al. \(2001\)](#) defendem que uma vantagem desta representação é permitir indexação com respeito a outras distâncias, além da euclidiana, o que não é possível com outras representações clássicas, como DFT e *wavelets*. [Yi e Faloutsos \(2000\)](#) demonstram como utilizar a representação PAA – embora não utilizem este termo – para indexar séries com base em qualquer norma L_p . Entretanto, na classificação de séries temporais, esta representação é provavelmente mais conhecida por ser utilizada como passo intermediário para se obter a representação *Symbolic Aggregate approXimation* ([LIN et al., 2007](#)).

5.4 Representação SAX (*Symbolic Aggregate approXimation*)

SAX (*Symbolic Aggregate approXimation*) é uma representação simbólica que discretiza uma série temporal em uma cadeia de caracteres (*string*). A associação de representações simbólicas a séries temporais é um tópico de interesse porque permite que diversas técnicas de processamento de *strings* sejam empregadas em processamento de sinais. Embora não seja a única representação simbólica para séries temporais, esta foi a primeira representação simbólica a promover redução de dimensionalidade e simultaneamente garantir o *lower bounding* da distância euclidiana ([LIN et al., 2007](#); [WANG et al., 2013](#)).

O domínio SAX é definido por um alfabeto $\Sigma = \{w_1, w_2, \dots, w_N\}$. Em sua proposição original, a representação SAX de uma série temporal é uma palavra $w_{c_1}w_{c_2}\dots w_{c_a}$. Cada símbolo w_{c_k} é obtido pela discretização da representação PAA da série original. A discretização é definida de modo que a ocorrência de cada símbolo na palavra seja aproximadamente equiprovável. Para isso, os autores assumem que as observações da série, no domínio temporal, seguem aproximadamente a distribuição normal padrão e definem **pontos de corte** ao longo dessa distribuição, estabelecendo intervalos que podem ser associados a cada símbolo do alfabeto.

Definição 5.7: *Pontos de corte são uma lista de números $B = \beta_1, \beta_2, \dots, \beta_{a-1}$, tais que a área sob uma curva Gaussiana com distribuição $\mathcal{N}(0, 1)$ é $1/a$ nos intervalos estabelecidos por cada par β_i e β_{i+1} ([LIN et al., 2007](#)). Os intervalos abertos não limitados por dois pontos de corte são estabelecidos assumindo-se $\beta_0 = -\infty$ e $\beta_a = \infty$.*

A transformação SAX possui dois parâmetros. O parâmetro N define o número de coeficientes a serem utilizados na representação intermediária. O parâmetro a é o tamanho do alfabeto Σ . Esse parâmetro determina a “resolução” da série no domínio SAX. Quanto menos símbolos forem empregados, maior a probabilidade de que séries próximas no domínio temporal sejam representadas pela mesma palavra no domínio SAX. A representação SAX com base em um alfabeto vazio não pode ser definida, enquanto um alfabeto com um único símbolo produziria palavras indistinguíveis e não teria utilidade. Um alfabeto com dois símbolos estabeleceria um

único ponto de corte $\beta_1 = 0$. Para alfabetos de tamanhos entre 3 e 12, seus pontos de corte podem ser consultados Tabela 12.

Tabela 12 – Pontos de corte para a representação SAX

a	β_1	β_2	β_3	β_4	β_5	β_6	β_7	β_8	β_9	β_{10}	β_{11}
3	-0,43	0,43									
4	-0,67	0	0,67								
5	-0,84	-0,25	0,25	0,84							
6	-0,97	-0,43	0	0,43	0,97						
7	-1,07	-0,57	-0,18	0,18	0,57	1,07					
8	-1,15	-0,67	-0,32	0	0,32	0,67	1,15				
9	-1,22	-0,76	-0,43	-0,14	0,14	0,43	0,76	1,22			
10	-1,28	-0,84	-0,52	-0,25	0	0,25	0,52	0,84	1,28		
11	-1,34	-0,91	-0,60	-0,35	-0,11	0,11	0,35	0,60	0,91	1,34	
12	-1,38	-0,97	-0,67	-0,43	-0,21	0	0,21	0,43	0,67	0,97	1,38

É possível encontrar pontos de cortes para alfabetos com $a > 12$. Entretanto, observa-se empiricamente que alfabetos excessivamente grandes produzem resultados pobres em tarefas de classificação. Portanto o estudo conduzido neste trabalho limita-se a um alfabeto com $a \leq 12$.

Definido o parâmetro a , a relação entre pontos de cortes e símbolos SAX dá-se como o seguinte.

Definição 5.8: Uma observação ou coeficiente $x_i \in \mathbb{R}$ é mapeada para um símbolo $\rho(x_i) = w_j$ se a condição $\beta_{j-1} \leq x_i < \beta_j$ é satisfeita.

Para se obter a distância entre duas palavras no domínio SAX, os autores propõem uma função de distância denominada MINDIST, a qual estabelece uma ordem entre as “letras” da palavra SAX. Nesta Tese, a fim de permitir que esta representação possa ser utilizada com as mesmas funções de distância que se aplicam no domínio temporal, o alfabeto é mapeado para o intervalo $[1, a]$ e a **representação SAX** é definida da seguinte maneira.

Definição 5.9: Dada uma série temporal $S = (s[1], s[2], \dots, s[n])$, cuja representação PAA é a sequência $PAA(S) \equiv (x_1, x_2, \dots, x_N)$ e um alfabeto $\Sigma = \{w_1, w_2, \dots, w_a\}$, a representação SAX de S é

$$SAX(S) = (c_1, c_2, \dots, c_N), \quad (5.16)$$

no qual cada c_i é tal que $\rho(x_i) = w_{c_i}$.

Ou seja, a representação SAX de uma série temporal é o conjunto ordenado dos índices dos símbolos que compõem sua palavra SAX. Um exemplo desse procedimento é mostrado na Figura 47.

Todos os elementos relevantes a esta representação são mostrados na Figura 47: a curva junto ao eixo das ordenadas representa a distribuição Gaussiana $\mathcal{N}(0, 1)$. As linhas horizontais, definidas implicitamente pelo alfabeto de tamanho $a = 3$, são posicionadas nas coordenadas β_1 e

Figura 47 – Representação de uma série no domínio SAX – exemplo extraído de (LIN *et al.*, 2007). A série é inicialmente transformada para o domínio PAA com $N = 8$. Em seguida, dois pontos de corte são estabelecidos e três faixas de discretização são associadas aos símbolos a , b e c . A palavra SAX dessa série é a sequência $baabccbc$ e sua representação SAX é $(2, 1, 1, 2, 3, 3, 2, 3)$.

β_2 . A série temporal original é mostrada como a curva ao fundo da imagem. Em primeiro plano, mostra-se a representação PAA dessa série obtida com $N = 8$. A palavra SAX associada a essa série seria, portanto, $baabccbc$ e sua representação SAX seria $(2, 1, 1, 2, 3, 3, 2, 3)$.

A MINDIST é definida da seguinte forma:

Definição 5.10: Sejam $Q = (q_1, q_2, \dots, q_N)$ e $C = (c_1, c_2, \dots, c_N)$ duas palavras SAX de tamanho N sobre um alfabeto de tamanho a $\Sigma = \{w_1, w_2, \dots, w_a\}$. A distância MINDIST entre Q e C é

$$MINDIST(Q, C) = \sqrt{\frac{n}{N}} \sqrt{\sum_{i=1}^N (d_L(q_i, c_i))^2}, \quad (5.17)$$

na qual n é o tamanho original das séries e d_L é a distância lookup entre dois símbolos do alfabeto SAX $q_i = w_q$ e $c_i = w_c$:

$$d_L(w_q, w_c) = \begin{cases} 0, & \text{se } |q - c| \leq 1 \\ \beta_{(\max(q,c)-1)} - \beta_{\min(q,c)}, & \text{caso contrário} \end{cases} \quad (5.18)$$

A relevância da MINDIST é o fato de que ela permite *lower bounding* da distância euclidiana na representação PAA que, por sua vez, é um *lower bounding* da distância euclidiana na representação temporal. Em trabalhos nos quais a propriedade do *lower bounding* é importante (e.g., indexação), pode-se utilizar a MINDIST para comparar séries na representação SAX. Para as aplicações de classificação consideradas nesta Tese, as funções de distância apresentadas no Capítulo 4 são empregadas.

Uma aplicação da representação SAX é detecção de anomalias. Essas séries devem ser caracterizadas por uma distância elevada a seus vizinhos mais próximos. Qualquer outro par de séries mais próximos do que as séries de referência não são anomalias e podem ser descartadas do processo de busca. Keogh, Lin e Fu (2005) utilizam a representação SAX (Seção 5.4) e selecionam as séries que são mapeadas às palavras SAX menos frequentes para utilizar como conjunto de referência.

5.5 Componentes Principais

A **análise das componentes principais** ou a **PCA** (*principal component analysis*) é uma técnica de estatística descritiva que permite identificar padrões de covariância em uma amostra dos dados (JOLLIFFE, 2002). Trata-se de uma transformação ortonormal que permite representar um conjunto de variáveis potencialmente correlacionadas como um conjunto de variáveis cuja correlação (linear) é mínima. É utilizada muitas vezes como uma técnica de redução de dimensionalidade. Também é conhecida como SVD (*singular value decomposition*).

A transformação PCA de um conjunto de dados pode ser realizada de forma iterativa. Os dados são representados como um conjunto de pontos $D \in \mathbb{R}^n$. A primeira componente principal é um vetor u_1 tal que a projeção de todos os pontos sobre esse vetor resulta na máxima variância possível. Seja S a matriz de covariância do conjunto D e $f(u_1)$ a variância da projeção de D sobre u_1 . Então, podemos encontrar u_1 por meio do problema de otimização

$$(\max) \quad f(u_1) = u_1^T S u_1 \quad (5.19)$$

$$\text{s/a} \quad g(x) = 1 - u_1^T u_1. \quad (5.20)$$

Note que u_1^T significa a transposta do vetor u_1 . A condição na Equação 5.20 assegura que o vetor u_1 é unitário e que a transformação obtida será ortonormal. Para encontrar as demais componentes principais, encontra-se o próximo vetor cuja projeção dos dados produz a maior variância possível, dado que esse vetor é ortogonal aos que foram obtidos nos passos anteriores (TAN; STEINBACH; KUMAR, 2006).

Uma ilustração desse processo é feita na Figura 48. Na Figura 48-esquerda, um conjunto de dados descritos por dois atributos, x e y , é representado como pontos no plano cartesiano. Os vetores ortogonais destacados sobre os pontos indicam as componentes principais e onde ocorre a maior variância dos dados. Podemos tornar essa relação explícita construindo uma base com esses vetores, como mostrado na Figura 48-direita.

Figura 48 – Exemplo de transformação PCA. Um conjunto de pontos descrito pelos atributos x e y no qual se identifica dois vetores ortogonais ao longo dos quais se observa maior variância dos dados (*esquerda*) é transformado para um conjunto de pontos descrito pelos atributos x' e y' , de modo que a maior variância dos dados encontra-se ao longo das novas dimensões (*direita*) – adaptado de (TAN; STEINBACH; KUMAR, 2006)

A transformação PCA normalmente é utilizada para encontrar correlações inesperadas entre as variáveis ou para promover redução de dimensionalidade. A variância é máxima na direção da primeira componente principal e reduz assintoticamente conforme o número de componentes extraídas, de modo que a quantidade de informação redundante aumenta com as componentes de maior ordem. Neste trabalho, utiliza-se a PCA como representação na qual funções de distâncias alternativas à euclidiana podem ser utilizada para explorar correlações lineares entre as observações.

Definição 5.11: *Seja $D = \{S_1, S_2, \dots, S_N\}$ um conjunto de séries de treinamento de dimensão $|S_i| = n$. Seja $U = \{u_1, u_2, \dots, u_n\}$ uma base ortonormal formada pelas componentes principais $u_{1,2,\dots,n}$ da matriz de dados formada pelas observações de D . Seja o vetor*

$$S_i U = (S_i u_1, S_i u_2, \dots, S_i u_n) \quad (5.21)$$

a projeção das observações da série S_i sobre as componentes principais, então a representação PCA de uma série temporal S_i é dada por $PCA_U(S_i)$.

$$PCA_U(S_i) = S_i U \quad (5.22)$$

Note que, diferentemente das demais representações, a PCA necessita de um contexto no qual existe um conjunto de séries de referência. Para se obter a representação PCA de uma série de teste ou validação, é necessário conhecer a base U associada aos exemplos de treinamento.

5.6 Representação autocorrelativa

Dado um processo estocástico $\mathbf{X} = \{\mathbf{X}_1, \mathbf{X}_2, \dots\}$, tal que cada variável $\mathbf{X}_t$ tem média $\mu_{\mathbf{X}}(t)$ e covariância $\gamma_{\mathbf{X}}(r, s)$,

$$\mu_{\mathbf{X}}(t) = E[\mathbf{X}_t] \quad (5.23)$$

$$\gamma_{\mathbf{X}}(r, s) = \text{Cov}(\mathbf{X}_r, \mathbf{X}_s) \quad (5.24)$$

$$= E[(\mathbf{X}_r - \mu_{\mathbf{X}}(r))(\mathbf{X}_s - \mu_{\mathbf{X}}(s))], \quad (5.25)$$

diz-se que $\{\mathbf{X}_t\}$ é **estacionário** se $\mu_{\mathbf{X}}(t)$ é independente de t e $\gamma_{\mathbf{X}}(t+h, t)$ é independente de t para cada h (BROCKWELL; DAVIS, 2002). De certa forma, um processo estacionário é caracterizado por um mecanismo gerador que não se altera durante sua realização. A estacionariedade de um processo é normalmente assumida, por exemplo, em previsão de séries temporais, principalmente quando se utiliza abordagens baseadas em modelos como ARIMA (PARMEZAN, 2016).

Uma importante estatística que pode ser obtida de um processo estacionário é a autocorrelação. Ela é baseada na autocovariância do processo, que é uma medida da variância conjunta para diferentes lags h ,

$$\gamma_{\mathbf{X}}(h) = \text{Cov}(\mathbf{X}_{t+h}, \mathbf{X}_t). \quad (5.26)$$

Normalizando-se a autocovariância pela variância do processo ($\text{lag } h = 0$), obtém-se sua **autocorrelação**,

$$\rho_{\mathbf{X}}(h) \equiv \frac{\gamma_{\mathbf{X}}(h)}{\gamma_{\mathbf{X}}(0)}, \quad (5.27)$$

que é uma informação da “persistência” do processo. Padrões de autocorrelação podem indicar, por exemplo, a existência de ciclos mesmo na presença de ruído.

Note que a autocorrelação é obtida com base em informações do processo, que é o mecanismo por trás do fenômeno, em vez do fenômeno em si—*i.e.*, a série observada. Em aplicações práticas, os parâmetros do processo são normalmente desconhecidos. Nesse caso, utiliza-se valores amostrados das séries para estimar a autocorrelação.

Definição 5.12: *Seja $S = (s[1], s[2], \dots, s[n])$ uma série temporal com n observações e cujo valor médio é m (Equação 2.5, pág. 38). O valor amostrado da função de autocovariância de S é*

$$\hat{\gamma}(h) := \frac{1}{n} \sum_{t=1}^{n-h} (s[t+h] - m)(s[t] - m) \quad (5.28)$$

e a função de autocorrelação é

$$\hat{\rho}(h) := \frac{\hat{\gamma}(h)}{\hat{\gamma}(0)}, \quad (5.29)$$

para valores de $\text{lag } h \geq 0$. Note que, $\hat{\gamma}(0)$ é a variância da série temporal e, para séries temporais z -normalizadas, $\hat{\rho}(h) = \hat{\gamma}(h)$.

A função de autocorrelação é bastante utilizada para estimar parâmetros de modelos estatísticos como ARIMA. Embora a Definição 5.12 assuma estacionaridade, a função de autocorrelação pode ser estimada para qualquer série temporal. De fato, a autocorrelação pode ser utilizada para detectar se o processo é estacionário. Os coeficientes de autocorrelação de um processo estacionário rapidamente se aproximam de zero. Contraste o gráfico de autocorrelação de um processo estacionário, mostrado na Figura 49-esquerda, com a de um processo não estacionário, mostrado na Figura 49-direita.

Figura 49 – Gráfico de autocorrelação para um processo estacionário e um não estacionário.

Nesta Tese, a **representação autocorrelativa** ou **ACF** (*Autocorrelation Function*) é um domínio de representação no qual a série temporal é representada pela variação da “persistência” do fenômeno gerado em diferentes *lags*.

Definição 5.13: Seja $S = (s[1], s[2], \dots, s[n])$ uma série temporal com n observações, então sua representação ACF é

$$ACF(S) = (\hat{\rho}(0), \hat{\rho}(1), \dots, \hat{\rho}(n-1)), \quad (5.30)$$

tal que $\hat{\rho}(h)$ é a autocorrelação amostral de S no lag h .

O domínio de representação ACF pode ser particularmente eficaz em casos nos quais as classes são caracterizadas por diferentes padrões de persistência.

5.7 Atributos de distância

Considerando-se o esforço técnico e científico investido na pesquisa em técnicas para classificar séries temporais, o alto desempenho obtido pelo classificador 1-NN, em termos de acurácia de classificação (DING *et al.*, 2008; BATISTA; WANG; KEOGH, 2011; BAGNALL *et al.*, 2015), é surpreendente. O k -NN é um algoritmo relativamente simples que vem sendo empregado há pelo menos meio século (COVER; HART, 1967). Por que modelos de classificação mais complexos são frequentemente “vencidos” pelo k vizinhos mais próximos?

Uma possível explicação para a vantagem do k -NN é a dificuldade em se obter atributos temporais que possam ser explorados adequadamente por algum modelo de classificação mais complexo. Atributos devem ser extraídos de subsequências ou séries inteiras e não há uma estratégia padronizada para determinar quais subsequências devem ser utilizadas na extração de atributos. O classificador 1-NN, por outro lado, toma proveito de um espaço de decisão no qual uma função de distância promove uma comparação entre séries que envolve todas as observações e respeita a ordem temporal entre elas.

Atributos de distância são uma forma de utilizar o poder discriminativo das representações e funções de distância como atributos genéricos que podem ser utilizados na indução de diferentes modelos de classificação (GIUSTI; SILVA; BATISTA, 2016). Os atributos de distância estendem as *DTW features* (KATE, 2016) e estabelecem um espaço de decisão no qual as séries são representadas como vetores de distância com respeito a um conjunto de referência.

Um conjunto de atributos de distância para uma série S é extraído com respeito a uma representação R , uma função de distância f e um conjunto de séries temporais de referência $P = \{P_1, P_2, \dots, P_N\}$. Seja $\phi_R(S)$ a representação da série temporal S no domínio R . Então os atributos de distância de S são o vetor $S^{R,f} = (g(1), g(2), \dots, g(N))$, no qual adota-se a notação $g(i) := f(\phi_R(S), \phi_R(P_i))$ por conveniência. Por exemplo, atributos extraídos da representação espectral utilizando distância euclidiana são dados pela função de transformação $\phi_{FS}(S) = FS(S)$ (Equação 5.5) e pela função de distância $f(S, Z) = d_{Euc}(S, Z)$. Formalmente, o procedimento para encontrar os atributos de distância de uma série é simplesmente o Algoritmo 5.1.

Atributos de distância pode ser extraídos em qualquer representação. A função identidade $\phi_I(S) = S$ pode ser utilizada para extrair atributos de distância no domínio temporal. Os atributos

Algoritmo 5.1: Extração de atributos de distância.

```

1 função AD = AtributosDeDistância ( $S, \phi, f, P$ )
  Parâmetros de Entrada:
     $S$ : uma série temporal de interesse
     $\phi$ : função de transformação
     $f$ : função de distância
     $P = \{P_1, P_2, \dots, P_N\}$ : conjunto de séries de referência
  Parâmetros de Saída:
    AD: vetor contendo os atributos de distância de  $S$ 
2  $S \leftarrow \phi(S)$ 
3 para cada  $P_i \in P$  faça
4   |    $AD(i) \leftarrow f(S, \phi(P_i))$ 
5 fim

```

DTW, originalmente propostos em (KATE, 2016), são obtidos quando se utiliza a função de transformação ϕ_I e a função de distância $f(S, Z) = d_{DTW}(S, Z)$. Atributos de distância, assim como os atributos DTW, podem ser combinados, bastando para isso concatenar os vetores que representam os atributos. Isto é, se $S^{R,f}$ e $S^{R,g}$ são atributos de distância extraídos na representação R com as funções de distância f e g , então o vetor $(S^{R,f}, S^{R,g})$ é um conjunto de atributos de distância que combina os dois atributos originais em um único espaço.

Nesta Tese, esses atributos são utilizados para gerar um modelo de máquina de vetores de suporte (SVM, *Support Vector Machines*). Uma máquina de vetores de suporte é um classificador baseado em instâncias que utiliza um *kernel trick* para encontrar um plano de separação máxima entre as classes. Esse conceito é representado na Figura 50. O espaço de decisão ilustrado na figura é determinado por dois atributos, X_1 e X_2 e os círculos representam instâncias de um problema de classificação binária. As retas são exemplos de hiperplanos que separam as classes no espaço de decisão. Entretanto, apenas uma delas representa um hiperplano com a maior margem de separação, que é a distância mínima entre o hiperplano e uma instância. As instâncias que coincidem com a margem ao hiperplano de separação ótima são chamadas vetores de suporte.

Para encontrar o hiperplano de separação ótima, o indutor de máquinas de vetores de suporte soluciona um problema de otimização quadrática (THEODORIDIS; KOUTROUMBAS, 2009). Esse problema é definido de forma tal que, em vez de minimizar alguma função de erro no conjunto de treinamento, o hiperplano maximiza uma margem de separação a fim de tentar minimizar o erro de generalização do classificador (RUSSELL; NORVIG, 2010). Para isso, o indutor utiliza um *kernel trick* que transporta os exemplos de treinamento para um espaço de maior dimensão, denominado espaço de características. Além disso, o SVM utiliza um limiar de tolerância que define uma margem relaxada, como ilustrado na Figura 51. A instância destacada em ambos espaços não pôde ser separada das instâncias da classe escura (vermelha), no espaço de características definido pelo *kernel*.

Um importante resultado do *kernel trick* é que o espaço de características não precisa

Figura 50 – Ilustração do modelo de máquina de vetores de suporte (PARMEZAN, 2016).

Figura 51 – Ilustração do *kernel trick*—adaptado de (PARMEZAN, 2016).

ser explicitamente encontrado pelo classificador. Os coeficientes dos exemplos no espaço de características não são conhecidos. Em vez disso, o *kernel* fornece a distância entre os exemplos nesse espaço. Isso permite o emprego de *kernels* extremamente complexos e até mesmo em espaços de dimensionalidade infinita. Para classificar um novo exemplo, o SVM encontra a distância dele para os vetores de suporte por meio do *kernel*, identificando assim o sub-espaço no qual o exemplo é projetado no espaço de características.

5.8 Avaliação experimental

Nesta seção, apresentam-se experimentos utilizando representações individualmente. A principal hipótese desta Tese é que nenhuma representação pode individualmente ser superior às demais em todo domínio de aplicação. Essa hipótese é avaliada na Seção 5.8.2.

Os resultados obtidos desse estudo foram empregados como base para os experimentos envolvendo *ensembles* de classificadores, os quais serão apresentados nos próximos capítulos desta Tese.

Além disso, são discutidos também nesta seção experimentos de atributos de distância. Os resultados apresentados na Seção 5.8.3 foram publicados em (GIUSTI; SILVA; BATISTA, 2016).

5.8.1 Ajuste de parâmetros

Com exceção da representação SAX, todas as demais representações estudadas neste trabalho possuem no máximo um parâmetro que necessita ser ajustado. A representação de *wavelets* possui como parâmetro a *wavelet* mãe, utilizada para produzir os filtros que definem os coeficientes de aproximação e de detalhe na transformada DWT. Nesta Tese, foram avaliados diferentes casos da *wavelet* de Daubechies. Os espaços de busca são descritos na Tabela 13.

Tabela 13 – Descrição dos espaços de busca dos parâmetros das representações.

Representação	Parâmetro	Descrição	Espaço de busca
DWT	–	<i>Wavelet</i>	Db1, Db2, ..., Db9
PAA	N	Número de segmentos	*
SAX	a	Tamanho do alfabeto	{2, 3, 4, 5, 7, 9, 12}
SAX	N	Número de segmentos	*
PCA	p	Número de componentes	10%, 20%, ..., 100% do número de observações
ACF	c	Número de coeficientes	10%, 20%, ..., 100% do número de observações

A representação FS não possui parâmetros. Para o número de segmentos nas representações PAA e SAX, os espaços de busca foram definidos de acordo com o tamanho das séries. Para conjuntos com até 10 observações, foram testados todos os valores no espaço $N = 2, 3, \dots, 10$. Para conjuntos com até 20 observações foram testados, além destes, o espaço $N = 12, 14, \dots, 20$ e, para conjuntos com mais que 20 observações, foram incluídos $N = 25, 30, 35, \dots, \lfloor \frac{n}{2} \rfloor$, de modo que cada segmento foi construído com pelo menos duas observações. A decisão por esse espaço de busca foi baseada na hipótese de que diferenças no parâmetro N são mais significativas quando N é pequeno.

Note que os valores ideais de cada parâmetro variam não só com os conjuntos de dados, mas também com as funções de distância utilizadas para encontrar os valores ideais. Como exemplo, apresenta-se na Tabela 14 uma amostra de valores de acurácias obtidas, acompanhadas dos valores ideais dos parâmetros para as representações PCA e ACF.

Tabela 14 – Melhores acurácias e parâmetros para um subgrupo de conjuntos nas representações PCA e ACF.

Conjunto	Melhores acurácias para PCA (parâmetro p)				
	Euclidiana	Manhattan	Hellinger	χ^2 Quadrada	
BeetleFly	0.7500 (102)	0.7000 (102)	0.6500 (102)	0.8000 (102)	
BirdChicken	0.5500 (102)	0.5500 (102)	0.6500 (102)	0.6000 (102)	
Coffee	1.0000 (114)	0.9643 (114)	0.9643 (29)	0.9643 (29)	
Beef	0.6667 (141)	0.7667 (141)	0.8667 (141)	0.7333 (47)	
OliveOil	0.8667 (114)	0.9000 (114)	0.8667 (114)	0.8667 (171)	
Wine	0.6111 (117)	0.6111 (117)	0.5370 (117)	0.6296 (117)	
FaceFour	0.7841 (105)	0.7841 (105)	0.7159 (105)	0.6250 (105)	
SonyAIBORobotSurface	0.6988 (14)	0.6822 (14)	0.7072 (14)	0.7238 (14)	
ECG200	0.9000 (10)	0.9000 (67)	0.8900 (10)	0.8600 (19)	
GunPoint	0.9133 (105)	0.9200 (105)	0.9533 (105)	0.9000 (30)	

Conjunto	Melhores acurácias para ACF (parâmetro c)				
	Euclidiana	Manhattan	Hellinger	χ^2 Quadrada	
BeetleFly	0.3500 (154)	0.4500 (154)	0.6000 (51)	0.6000 (51)	
BirdChicken	0.5500 (307)	0.4500 (307)	0.4000 (307)	0.4000 (256)	
Coffee	0.5357 (114)	0.4643 (114)	0.4643 (114)	0.4643 (114)	
Beef	0.2333 (470)	0.2000 (423)	0.3000 (329)	0.3333 (423)	
OliveOil	0.1333 (399)	0.1333 (399)	0.1333 (285)	0.1333 (399)	
Wine	0.5000 (140)	0.5000 (164)	0.5000 (164)	0.5000 (164)	
FaceFour	0.4659 (245)	0.3636 (245)	0.4659 (175)	0.1932 (315)	
SonyAIBORobotSurface	0.4309 (21)	0.4443 (21)	0.5141 (56)	0.5458 (42)	
ECG200	0.8200 (58)	0.8300 (48)	0.4200 (19)	0.3900 (19)	
GunPoint	0.4933 (135)	0.4933 (135)	0.4933 (105)	0.4933 (105)	

As configurações ideais foram encontradas para todos os conjuntos do repositório UCR, empregando um subconjunto das funções de distância apresentadas no capítulo anterior:

- Funções que não necessitam normalização: euclidiana, Manhattan, Chebyshev, média aritmética entre Manhattan e Chebyshev, Dice, Canberra, Lorentzian, Sørensen, cosseno e Jaccard;
- Funções que necessitam normalização para o intervalo $[0, 1]$: χ^2 de Pearson, χ^2 quadrada e Jeffrey;
- Funções que necessitam normalização como histogramas: Kullback-Leibler, Bhattacharyya e Hellinger.

Os valores resultantes de acurácia esperada e obtida são excessivos para serem reproduzidos neste texto, mas podem ser consultados em (GIUSTI, 2017). Na Tabela 15 são apresentadas as médias dos valores de acurácia obtida para os conjuntos do repositório UCR. Uma discussão sobre o emprego de diferentes distâncias associado a diversidade de representação será feita nas próximas subseções.

5.8.2 Representações individuais

A maneira mais “óbvia” de se empregar diversidade de representações na classificação de séries temporais é utilizar uma única representação alternativa no classificador 1-NN. Inicialmente, esse experimento foi conduzido com todos os classificadores utilizando a distância

Tabela 15 – Acurácias médias de distâncias e representações sobre todos os conjuntos do repositório UCR. Uma versão desta tabela mostrando apenas vitórias, empates e derrotas pode ser encontrada na pág. 164.

Distância	Representações						
	Tempo	FS	DWT	PAA	SAX	PCA	ACF
Média L_1/L_∞	0.7221	0.7042	0.7130	0.7314	0.7046	0.7176	0.7010
Bhattacharyya	0.6995	0.6962	0.6973	0.6445	0.6732	0.6970	0.6853
Canberra	0.6976	0.5835	0.7220	0.6937	0.7008	0.6569	0.6803
Chebyshev	0.6549	0.6718	0.6808	0.7026	0.6237	0.6966	0.6884
Cosseno	0.7123	0.7008	0.7072	0.7164	0.7025	0.7144	0.6933
Dice	0.7123	0.7008	0.7044	0.7206	0.7077	0.7147	0.6988
Euclidiana	0.7123	0.7008	0.7093	0.7206	0.6983	0.7168	0.6971
Hellinger	0.7013	0.6960	0.6969	0.4998	0.6762	0.6966	0.6849
Jaccard	0.7123	0.7008	0.7044	0.7206	0.7077	0.7147	0.6988
Jeffrey	0.6700	0.6961	0.6630	0.4914	0.6589	0.6526	0.6863
Kullback-Leibler	0.7065	0.6952	0.6952	0.4845	0.6840	0.6998	0.6852
Lorentzian	0.7244	0.6458	0.7095	0.7270	0.7032	0.7011	0.6974
Manhattan	0.7226	0.7060	0.7148	0.7251	0.7044	0.7154	0.7013
χ^2 Pearson	0.6974	0.6465	0.6642	0.4913	0.6958	0.6779	0.6701
Sørensen	0.7202	0.7066	0.7296	0.7282	0.7098	0.7205	0.6981
χ^2 Quadrada	0.7164	0.6984	0.6659	0.4906	0.6955	0.6887	0.6907

euclidiana. Os valores de acurácia para as seis representações alternativas apresentadas ao longo deste capítulo são mostrados no apêndice (Seção A.8, pág. 229). Para os domínios que exigem parâmetros, foram empregadas as configurações ideais selecionadas no experimento anterior.

Um teste de Friedman foi realizado. Com $\alpha = 0,95$, o valor crítico da estatística-F é 2,12 para sete algoritmos e 85 conjuntos. A estatística-F encontrada para o experimento foi 4,68. Na Figura 52, apresenta-se um gráfico de distância crítica.

Figura 52 – Gráfico CD para as representações com distância euclidiana.

Os resultados sugerem que as representações são estatisticamente indistinguíveis. Com exceção, possivelmente, da representação PCA que, no caso médio, produz classificadores melhores do que aqueles obtidos nas representações ACF e SAX. Podemos observar a diferença entre essas representações de forma mais clara por meio de gráficos de espalhamento (Figura 53).

Os resultados evidenciam que não é satisfatório depender de nenhuma representação específica para classificar séries temporais. A questão subsequente é se o desempenho médio de cada representação pode ser melhorado utilizando-se outras funções de distância. Na Tabela 16, apresenta-se um estudo do desempenho médio do classificador 1-NN quando utilizando diversidade de distância nas representações escolhidas. Nessa tabela, o símbolo ▲ indica que houve diferença estatisticamente significativa, de acordo com o teste de hipótese pareado de Wilcoxon, e que a função de distância da linha é melhor que a distância euclidiana. O símbolo ▼ indica o

Figura 53 – Gráficos de espalhamento para PCA contra ACF e SAX.

oposto. Os demais casos foram obtidos na ausência de diferença estatisticamente significativa.

Note que, mesmo no domínio temporal, apenas as distâncias baseadas na norma L_1 produziram classificadores superiores ao 1-NN com distância euclidiana. Esse resultado, que parece ir contra aquele que foi observado na Seção 4.7.2.2 serve como um lembrete de que se deve empregar cautela ao tomar conclusões baseadas no comportamento médio de um algoritmo.

Foi então conduzido um experimento para verificar o desempenho do 1-NN dinâmico sobre as representações alternativas. O procedimento adotado foi o mesmo da Seção 4.7.2.2, utilizando 16 funções de distância não elásticas. Os resultados desse experimento são muito longos para serem reproduzidos nesta Tese, mas podem ser consultados eletronicamente em (GIUSTI, 2017). Dois casos foram escolhidos e gráficos de espalhamento são mostrados na Figura 54.

Trata-se de um resultado encorajador, pois mostra que o esforço em selecionar funções

Tabela 16 – Testes de hipótese das distâncias contra a distância euclidiana

Distância	Representações						
	Tempo	FS	DWT	PAA	SAX	PCA	ACF
Média L_1/L_∞	▲	-	-	▲	▲	-	-
Bhattacharyya	▼	-	▼	▼	▼	▼	▼
Canberra	▼	▼	-	▼	-	▼	▼
Chebyshev	▼	▼	▼	▼	▼	▼	-
Cosseno	-	-	-	-	-	-	-
Dice	-	-	▼	-	-	-	-
Hellinger	-	-	▼	▼	▼	▼	▼
Jaccard	-	-	▼	-	-	-	-
Jeffrey	▼	-	▼	▼	▼	▼	▼
Kullback-Leibler	-	-	▼	▼	▼	▼	▼
Lorentzian	-	▼	-	▲	-	▼	-
Manhattan	▲	-	-	▲	-	-	-
χ^2 Pearson	▼	▼	▼	▼	-	▼	▼
Sørensen	-	-	▲	-	▲	-	-
χ^2 Quadrada	-	-	▼	▼	-	▼	-

Figura 54 – Gráficos de espalhamento para seleção de distância contra distância euclidiana.

de distância é recompensador mesmo para representações diferentes da temporal. Esse resultado foi utilizado nos estágios posteriores da execução desse trabalho, principalmente, na construção de *ensembles* de representações.

5.8.2.1 Diversidade de representações e distâncias elásticas

A próxima questão ser respondida é se podemos associar distâncias elásticas a mudanças de representações. Para isso, é necessário repetir os procedimentos de ajuste de parâmetros visando encontrar não apenas as configurações ideais das funções de transformação, mas também das funções de distâncias. Como as funções de distância são computacionalmente caras, decidiu-se iniciar com um experimento preliminar, utilizando apenas as representações FS e ACF. A representação FS não possui parâmetros associados e optou-se por utilizar a totalidade dos coeficientes de autocorrelação na relação ACF.

Os espaços de busca para as melhores configurações foram os mesmos escolhidos para o domínio temporal (Tabela 9), com uma diferença: para o parâmetro δ nas distâncias DTW e LCSS, que indica a diferença máxima em número de observações para um casamento, foi acrescentado o espaço $\delta^+ = \{1, 2, \dots, 20\}$ observações.

Em cada caso, comparou-se a distância elástica contra a distância euclidiana no domínio temporal. Os experimentos foram conduzidos em 60 conjuntos do repositório UCR. De acordo com esse experimento, os melhores resultados foram obtidos quando se empregou a distância DTW no domínio FS, seguidos da distância DTW no domínio ACF. Os piores resultados foram obtidos com a distância LCSS no domínio ACF. Os resultados são muito extensos e podem ser consultados na página *web* desta tese (GIUSTI, 2017). Na Figura 55 são mostrados dois gráficos de espalhamento considerados representativos (melhor e pior casos).

Os resultados obtidos pela distância DTW no domínio ACF (Figura 55-esquerda) e no domínio FS (GIUSTI, 2017) foram considerados “razoáveis”. Entretanto, as outras funções de distância apresentaram baixo desempenho. Isso não significa, entretanto, que essas distâncias não

Figura 55 – Desempenho do 1-NN com distâncias elásticas nos domínios FS e ACF.

possam ser utilizadas em um classificador 1-NN dinâmico elástico, como foi feito com sucesso no domínio temporal. Mas deve-se levar em consideração se o esforço necessário para ajustar os parâmetros, tanto da função de distância quanto do domínio de representação—lembrando que é necessário percorrer o espaço de busca da função de distância *para cada configuração* da função de transformação. Considerando-se que apenas as distâncias não elásticas já promovem um classificador bastante adequado nas representações alternativas, optou-se limitar as distâncias elásticas ao domínio temporal nos experimentos deste trabalho.

5.8.3 Atributos de distância

Os experimentos relatados nesta seção foram realizados com um número reduzido de representações e publicados em (GIUSTI; SILVA; BATISTA, 2016). As representações escolhidas foram o domínio temporal, o espectro de frequências, a *wavelet* de Haar e a função de autocorrelação. Para a função de autocorrelação, foram utilizados todos os coeficientes nos cálculos da distância. Essa decisão, bem como a decisão de se empregar apenas a *wavelet* de Haar na representação DWT, teve como objetivo eliminar parâmetros associados às representações.

Foram empregadas oito funções de distância na extração dos atributos. As funções de distância escolhidas foram a distância euclidiana, Chebyshev, média L_1/L_∞ , Jaccard, Dice e Lorentzian. Além disso, foram extraídos atributos de distância no domínio temporal associados à distância DTW com janela Sakoe-Chiba e sem mecanismo de janela.

Os classificadores foram induzidos com *kernel* polinomial através do *software* Weka (HALL *et al.*, 2009) e a biblioteca LibSVM (CHANG; LIN, 2011). O *kernel* polinomial expressa a distância entre os vetores S e Z como a relação

$$\mathbf{K}(S, Z) = (\gamma S^T \cdot Z + C)^d, \quad (5.31)$$

na qual S^T é o o vetor S transposto. O produto $S^T \cdot Z$ tem relação direta com a distância entre as séries no espaço de características. Os hiperparâmetros γ and C são empregados para ajustar o

kernel: valores mais elevados de γ significam que as diferenças entre os vetores no espaço de entrada serão mais significativas no espaço de características, enquanto o hiperparâmetro C tem efeito oposto. Neste experimento, foi observado o mesmo procedimento adotado por Kate (2016) e foram mantidos os valores originais de γ e C da biblioteca LibSVM Chang e Lin (2011)².

O parâmetro d estabelece o grau do polinômio utilizado no *kernel*. Seu valor está associado ao viés do SVM. Um polinômio de maior grau permite ao indutor encontrar hipóteses mais complexas, mas que podem estar super-adaptadas aos dados e não generalizar bem para exemplos novos. Por outro lado, um polinômio de grau subestimado pode não ser capaz de capturar o conceito necessário para realizar a classificação. O valor adequado para esse hiperparâmetro foi escolhido, para cada caso, por meio de validação cruzada. Foram explorados polinômios lineares e de segundo e terceiro graus.

No restante deste texto, será adotada a notação REPRESENTAÇÃO-DISTÂNCIA para identificar atributos de distância. Assim, o termo FS-MAN refere-se a atributos de distância extraídos sobre a representação espectral utilizando a distância Manhattan. Para atributos extraídos no domínio temporal, relata-se apenas o nome da distância. Os atributos referentes à distância DTW com janela foram denominados DTWR (“*warping* restrito”) e os referentes à distância DTW sem restrição de *warping* foram denominados DTW. Uma descrição do tipo EUC+PS-EUC indica composição dos atributos EUC e PS-EUC.

Para cada conjunto de dados, foram extraídos atributos de distância combinando as quatro representações e as seis funções de distância escolhidas para este experimento, totalizando 24 configurações por conjunto de dados. O polinômio foi ajustado para cada configuração através de validação cruzada 5×2 . Os resultados são excessivos para serem listados na Tese, mas podem ser consultados no endereço eletrônico indicado em (GIUSTI; SILVA; BATISTA, 2016)³. Para cada conjunto de dados, foi selecionada a configuração de maior acurácia esperada. O conjunto de atributos resultante desse procedimento foi denominado DISTF-tuned. Esse classificador foi comparado com o classificador 1-NN utilizando distância DTW e um gráfico de espalhamento é mostrado na Figura 56.

O gráfico é sugestivo de que os atributos de distância podem ser utilizados para produzir classificadores com maior acurácia que o 1-NN utilizando distância DTW. Esse resultado foi confirmado através de um teste de hipótese de Wilcoxon.

Foram também extraídos atributos no domínio temporal utilizando as duas variantes da distância DTW, produzindo o atributo composto DTW+DTWR. Esse atributo de distância é exatamente o conjunto de DTW-Features proposto por Kate (2016). O atributo de distância DTW também foi associado ao DISTF-tuned, produzindo o espaço de atributos composto DTW+DISTF-tuned. Em ambos os casos, o grau do polinômio foi ajustado nos exemplos de treinamento compostos através de 5×2 CV.

² A biblioteca LibSVM usa, por padrão, os valores $\gamma = \frac{1}{|D|}$ and $C = 1$.

³ <<http://sites.labc.icmc.usp.br/rg/icmla16/>>

Figura 56 – Comparação de atributos de distância com o 1-NN DTW.

Os testes seguintes foram realizados com o objetivo de comparar o classificador SVM produzido pelo atributo de distância composto DTW+DISTF-tuned com o 1-NN DTW e também com o classificador SVM utilizando DTW-Features. Gráficos de espalhamento são apresentados, respectivamente, na Figura 57-esquerda e Figura 57-direita.

Figura 57 – Comparação de atributos de distância e DTW com classificadores selecionados.

Novamente, os resultados sugerem que o classificador composto por atributos de distância com a distância DTW é superior em ambos os casos. Esses resultados foram confirmados através de testes de Wilcoxon.

Finalmente, foi conduzido um teste de hipótese de Friedman com os quatro métodos avaliados neste experimento: DTW+DISTF-tuned, DISTF-tuned, DTW+DTWR e 1-NN DTW. A hipótese nula de que os classificadores são idênticos foi rejeitada e um teste de Nemenyi foi conduzido em caráter post-hoc, encontrando a distância crítica dos *ranks* $CD = 0,51$. Um gráfico CD é apresentado na Figura 58.

Figura 58 – Gráfico de distância crítica para os métodos de classificação avaliados neste experimento. A diferença crítica de *ranks* médios é 0,51.

5.9 Considerações finais

Diversidade de representações é uma importante ferramenta para a condução de várias tarefas de análise de séries temporais. Neste capítulo, foram estudadas algumas representações relevantes para tarefas como classificação, indexação e segmentação de séries temporais, com ênfase na classificação.

Trabalhou-se aqui com a hipótese de que nenhuma representação pode evidenciar as características discriminativas das classes para todos os domínios de aplicação. Uma consequência dessa hipótese é a constatação de que é possível encontrar, para algum domínio de aplicação específico, a representação mais adequada de um conjunto de representações conhecidas. Os experimentos apresentados neste capítulo relatam o trabalho conduzido, ao longo deste projeto, para buscar formas apropriadas de se atingir esse fim.

Foi mostrado, na Seção 5.8.2, que é possível selecionar a função de distância mais apropriada para cada representação. Com isso, foi possível obter um classificador muito mais preciso do que seria possível caso apenas a distância euclidiana fosse empregada. Um resultado inesperado foi o baixo desempenho das distâncias elásticas nos domínios de representação alternativos. Esperava-se que o mesmo ganho de acurácia obtido quando se emprega invariância a *warping* no domínio temporal fosse observado igualmente em outras representações. Entretanto, o que se observou foi que o procedimento de selecionar distâncias não elásticas tende a produzir resultados muito melhores com menor custo computacional do que o procedimento necessário para ajustar os parâmetros das distâncias elásticas. Essa informação foi de grande importância para a condução dos experimentos seguintes neste trabalho, utilizando *ensembles* de representação.

ENSEMBLES DE CLASSIFICADORES

6.1 Considerações iniciais

N^O APRENDIZADO INDUTIVO, um classificador é uma hipótese $h \approx \zeta$ que aproxima uma função-conceito. O indutor é um algoritmo que utiliza um viés para estabelecer um espaço de hipóteses factíveis H e que se utiliza de método de busca para encontrar a melhor aproximação $h \in H$ de acordo com algum critério de qualidade (MITCHELL, 1997). Um dos aspectos mais importantes no aprendizado de máquina é encontrar algoritmos com viés adequado para alguma aplicação de interesse. Como exemplificado nos capítulos anteriores, o desempenho de um classificador—em particular, para esta Tese, o 1-NN—pode variar drasticamente dependendo da função de distância, representação dos dados ou configuração de parâmetros utilizados para definir o espaço de decisão.

Uma forma de se produzir classificadores com maior acurácia é utilizar mais de uma hipótese na descrição do modelo. A combinação de modelos com diferentes tipos de viés pode reduzir a probabilidade de classificação incorreta. O classificador ótimo de Bayes é um conjunto de *todas* as possíveis hipóteses para um domínio de aplicação. Embora seja um modelo idealizado e impraticável mesmo para espaços de decisão muito reduzidos, existem outros modelos de composição de hipóteses que podem produzir classificadores com menor erro médio. Esses modelos de classificação podem ser coletivamente denominados *ensembles* de classificadores.

Existem diversas formas de se construir *ensembles* de classificadores. Nesta Tese, *ensembles* são utilizados principalmente para combinar diferentes representações de séries temporais em um espaço de decisão diversificado. A utilização de *ensembles* baseados em funções de distância não elásticas também foi explorada nesta Tese como uma alternativa de menor custo computacional a *ensembles* baseados em distâncias elásticas e mudança de representação.

Uma revisão geral da bibliografia em torno de *ensembles* de classificadores é apresentada na Seção 6.2. Nas seções 6.3 e 6.4 são discutidos os mecanismos de construção de *ensembles*.

Resultados experimentais são apresentados na Seção 6.5.

6.2 Ensembles de classificadores

Um *ensemble de classificadores* é um modelo de classificação misto no qual várias hipóteses são empregadas para decidir um novo caso (ALPAYDIN, 2010; BERNARDINI, 2006). O objetivo do aprendizado com *ensembles* é reduzir o risco de erro de classificação através da associação de diferentes tipos de viés. Por exemplo, considere um simples *ensemble* que combina cinco hipóteses em um esquema de classificação majoritária. Para que um exemplo seja incorretamente classificado pelo conjunto, é necessário que pelo menos três hipóteses o classifiquem incorretamente.

Os classificadores que compõem um *ensemble* são denominados **classificadores base**. Embora o *ensemble* seja em si um modelo de classificação, ele não necessariamente reside no mesmo espaço de hipóteses que os classificadores base. Isso pode significar que o *ensemble* é capaz de representar conceitos que os classificadores, individualmente, não conseguem aproximar de forma adequada. *Ensembles* de classificadores são frequentemente mais eficazes do que cada um de seus componentes isolados. Para isso é necessário que ele seja composto por classificadores diversos e individualmente precisos.

Um novo caso é classificado por um *ensemble* através de uma estratégia de composição de votos. Um **voto** nada mais é do que o rótulo atribuído por um classificador base a um exemplo. Uma estratégia de composição é um conjunto de preferências estabelecidas para escolher uma classe diante dos votos apresentados pelos classificadores base. Por exemplo, o voto majoritário é obtido quando o *ensemble* classifica um exemplo como a classe que recebe o maior número de votos.

Definição 6.1: *Um ensemble de classificadores é um conjunto de modelos de classificação $\{h_1, h_2, \dots, h_L\}$ sobre um mesmo espaço de classes $\{C_1, C_2, \dots, C_k\}$. Ao classificar um novo exemplo x , cada rótulo obtido $h_i(x) = C_{h_i}$ é um voto para a classe C_{h_i} . A classe atribuída pelo ensemble ao exemplo é encontrada por uma função de composição de votos.*

Para problemas de classificação binária, o *ensemble* pode ser considerado uma realização de um processo aleatório no qual cada classificador base possui probabilidade p_i de classificar corretamente um exemplo x . Assumindo-se independência entre os classificadores e que o classificador menos preciso tem probabilidade de acerto $p = \min(p_i)$, então a probabilidade de que pelo menos l classificadores encontrem a classe correta de um exemplo segue uma distribuição binomial e pode ser expressa como na Equação 6.1.

$$P(Z = l) = \binom{L}{l} p^l (1 - p)^{L-l} \quad (6.1)$$

Se $p > 0,5$, então a probabilidade de erro do *ensemble* diminui conforme o número de classificadores aumenta. Na Figura 59 são mostradas curvas de erros para *ensembles* nos quais os classificadores base possuem probabilidades de acerto $p_1 = 0,7$, $p_2 = 0,6$ e $p_3 = 0,55$.

Figura 59 – Erro do *ensemble* em função do número de classificadores base (BERNARDINI, 2006).

Em aplicações reais, normalmente é impossível garantir a independência dos erros cometidos pelos classificadores base. Os dados disponíveis para treinar os classificadores são limitados, bem como as configurações que podem ser associadas a um algoritmo de indução. Quanto maior o número de classificadores, menos plausível se torna a hipótese da independência. Mesmo assim, a tendência de um *ensemble* é se tornar mais fiel à função-conceito quando expandido com classificadores adequados.

Rokach (2010) sugere um *framework* para construção de *ensembles* que considera quatro componentes:

Conjunto de treinamento: um conjunto de exemplos rotulados D ;

Indutor base: um algoritmo I que capaz de produzir um classificador dado um conjunto de exemplos de treinamento $D_i \subseteq D$. Note que esse algoritmo não necessariamente deve produzir todos os classificadores utilizando a mesma linguagem de descrição;

Gerador de diversidade: os classificadores base devem ser diversos a fim de que os erros sejam aproximadamente independentes. A diversidade pode ser associada ao indutor base ou a um mecanismo de reamostragem dos exemplos rotulados D ;

Estratégia de voto: um algoritmo que combina todos os votos em uma única classe.

6.3 Estratégias de geração de diversidade

Ainda que normalmente não seja possível garantir independência dos erros cometidos pelos classificadores base, pode-se tentar aproximar esse resultado através de um conjunto diversificado de classificadores base. Essencialmente, existem três formas de se introduzir diversidade a um *ensemble*:

1. Através de reamostragem dos exemplos de treinamento, produzindo distribuições $D_1, D_2, \dots, D_L$ que serão utilizadas para induzir os classificadores base $h_1, \dots, h_L$;
2. Utilizando-se diferentes configurações de parâmetros para cada classificador base;
3. Empregando-se diferentes algoritmos de indução ou diferentes algoritmos de classificação.

Algumas estratégias bem conhecidas são baseadas fundamentalmente em uma ou outra dessas premissas. A estratégia de *bagging*, por exemplo, produz classificadores sobre diferentes subconjuntos dos exemplos de treinamento. O método AdaBoost (MITCHELL, 1997) associa pesos aos exemplos de treinamento. Ao produzir uma hipótese h_i para compor o *ensemble*, os pesos são ajustados de forma a priorizar os exemplos que o modelo h_i classificou incorretamente. Dessa forma, ao produzir o próximo classificador base, o indutor priorizará os exemplos que foram “negligenciados” por h_i .

Alguns *ensembles* para séries temporais podem ser encontrados na literatura. Um exemplo do segundo método de diversidade é encontrado em (OATES *et al.*, 2012). Nesse trabalho, os autores utilizam a representação *Bag of Patterns* (LIN; KHADE; LI, 2012) para extrair atributos de um conjunto de séries temporais. A representação *Bag of Patterns* é semelhante em princípio de funcionamento à representação saco-de-palavras, utilizada em mineração de textos: uma janela deslizante percorre as séries, extraíndo palavras SAX das subsequências. Essas palavras formam os padrões que são inseridos em um saco-de-padrões. O *ensemble* de (OATES *et al.*, 2012) é um conjunto de classificadores base induzidos com diferentes configurações de alfabeto e pontos de corte ao extrair as palavras SAX.

6.3.1 Bagging

O *bagging* (FLACH, 2012) (*bootstrap aggregating* ou *agregado bootstrap*) é uma estratégia bastante popular para construir *ensembles* de classificadores. O *bagging* é um *ensemble* baseado em reamostragem *bootstrap*. Tipicamente, todos os classificadores base desse tipo de *ensemble* são induzidos por um mesmo algoritmo. Sua diversidade deriva do fato que os classificadores base são induzidos sobre sub-amostras potencialmente distintas dos exemplos de treinamento.

Para construir um *ensemble* utilizando o procedimento *bagging*, realiza-se L iterações de reamostragem *bootstrap* do conjunto D . A cada iteração, obtém-se um multiconjunto D_i

contendo N exemplos de treinamento, os quais são utilizados para produzir uma hipótese h_i . Se necessário, pode-se empregar uma estratégia de validação no conjunto D_i para obter a acurácia esperada do classificador h_i —que poderá ser utilizada posteriormente na etapa de composição de votos—ou para ajustar parâmetros do indutor antes de induzir a versão definitiva do classificador base h_i .

Nesta Tese, os classificadores são normalmente variações do algoritmo 1-NN utilizando diversidade de representações e de funções de distância. Como o classificador 1-NN não possui um modelo explícito, os classificadores h_i são apenas os índices dos exemplos selecionados em cada etapa. O *ensemble* classifica um novo exemplo reconstruindo o subconjunto D_i utilizando os índices armazenados e encontrando seu vizinho mais próximo considerando apenas os exemplos em D_i .

Uma extensão bastante simples do *bagging*, denominada ***subspace sampling***, pode ser empregada para promover diversidade também no espaço de atributos. Para isso, é necessário que os exemplos de treinamento estejam associados a um conjunto de atributos $\{X_1, X_2 \dots, X_M\}$. Antes de cada iteração do *bagging*, seleciona-se aleatoriamente um subconjunto de $d < M$ atributos (sem repetição). No caso particular em que o *ensemble* é composto por árvores de decisão, esse tipo de modelo é conhecido como ***random forest***. Para séries temporais, é improdutivo tratar as observações individuais como se fossem atributos, mas [Deng et al. \(2013\)](#) propõem uma estratégia de seleção de subsequências aleatórias das quais são extraídos atributos como média, desvio padrão, obliquidade etc. Um conjunto de árvores de decisão induzido sobre esses atributos é conhecido como ***time series forest***.

6.3.2 Ensemble de representações

O principal objetivo deste trabalho é explorar o uso de diferentes representações de séries temporais como mecanismo para expor características discriminativas de um domínio. Podemos associar representações distintas em *ensemble* para obter *diversidade de representação*.

Nesta Tese, define-se um ***ensemble de representações*** como um modelo misto no qual os classificadores base podem estar associados a diferentes funções de mudança de representação. Mais especificamente, neste trabalho os classificadores que compõem um *ensemble* de representação são variantes do classificador 1-NN, cada qual associado a um domínio de representação. Para classificar um novo exemplo, ele deve ser primeiramente transformado para todas as representações envolvidas no *ensemble*. Em seguida, cada classificador base vota de acordo com a classe do vizinho mais próximo no espaço de decisão determinado pela representação a ele associada.

Similarmente, um *ensemble* de distâncias é aquele no qual os classificadores base são variantes do 1-NN, todas sobre o mesmo conjunto de exemplos no domínio temporal, porém cada qual empregando uma função de distância própria. [Lines e Bagnall \(2015\)](#) apresentam

um “*ensemble* elástico” no qual são utilizadas diversas funções de distância elásticas para classificar um novo exemplo. Um *ensemble* de representações também pode combinar diferentes funções de distância. Por exemplo, um *ensemble* pode ser construído com a distância DTW e a distância euclidiana no domínio temporal, associadas à distância euclidiana em outros domínios de representação. Essa configuração foi explorada durante este trabalho e produziu classificadores com excelente acurácia (GIUSTI; SILVA; BATISTA, 2015).

Como o classificador 1-NN não possui um modelo de conhecimento explícito, o *ensemble* de representações não possui uma fase de “treinamento”. Entretanto, ele possui uma fase de “configuração”, na qual todos os exemplos de treinamento são convertidos para os domínios de representação necessários. Além disso, como será mostrado na próxima seção, a maioria das estratégias de composição de votos utilizadas neste trabalho exige algum tipo de processamento dos exemplos de treinamento. Por exemplo, o voto majoritário composto exige avaliar o desempenho esperado de cada classificador base. Algumas representações e funções de distância são paramétricas e exigem validação para ajuste de parâmetros.

O trabalho mais semelhante ao desta Tese é um *ensemble* hierárquico baseado em transformações e diversidade de funções de distância (BAGNALL *et al.*, 2015). Um *ensemble* hierárquico é uma estrutura na qual os classificadores base podem ser modelos simples ou *ensembles*. O *ensemble* hierárquico apresentado nesse artigo é denominado “coletivo de *ensembles* de transformação” ou COTE (*collective of transformation ensembles*). O COTE combina *ensembles* que utilizam diversidade de representações, de funções de distância e de linguagens de representação. Por exemplo, o COTE envolve modelos de vizinhança (1-NN), *time series forests* e árvores de *shapelets*. Apesar de produzir classificadores bastante fiéis à função-conceito, a desvantagem do COTE é sua alta complexidade computacional, envolvendo 35 classificadores base, muitos dos quais requerem etapas custosas de ajuste de parâmetros.

6.4 Estratégias de composição de votos

As estratégias de composição de votos podem ser agrupadas em duas categorias: dinâmicas e estáticas. Estratégias estáticas definem critérios que permanecem constantes durante o uso de um *ensemble*. Por exemplo, o voto majoritário simples sempre seleciona a classe que recebe maior número de votos dos classificadores base. Estratégias dinâmicas podem empregar critérios que se adaptam às instâncias que desejamos classificar.

Neste trabalho, foram explorados mecanismos dinâmicos e estáticos derivados do voto majoritário. Além disso, foram avaliados mecanismos dinâmicos baseados em composição de *rankings*.

6.4.1 Voto majoritário

O voto majoritário é uma das estratégias mais simples de composição de votos, no qual cada classificador base contribui igualmente para a rotulação do novo exemplo. No caso ideal em que uma classe obtém maioria simples dos votos, o *ensemble* rotula o novo exemplo de acordo com essa classe. Desempates podem ser resolvidos com escolha aleatória das classes empatadas ou com o auxílio de um mecanismo de decisão post-hoc. Por exemplo, no caso do classificador 1-NN, pode-se favorecer o desempate ao classificador cuja distância do exemplo ao vizinho mais próximo seja a menor dentre os classificadores empatados.

Mais especificamente, esse tipo de composição de votos é denominado *voto majoritário simples*. Esse mecanismo pode ser facilmente estendido associando-se pesos aos classificadores base. No voto majoritário ponderado, cada classificador h_i vota em uma classe com peso w_i . O rótulo escolhido para um novo exemplo é aquele cuja classe está associada à maior soma de pesos.

Por exemplo, suponha um modelo misto de três classificadores binários h_1 , h_2 e h_3 , associados aos pesos $w_1=1$, $w_2=2$ e $w_3=4$. Suponha que, para uma instância x qualquer, os classificadores base produzem os rótulos $\oplus$, $\oplus$ e $\ominus$. De acordo com o voto majoritário simples, o rótulo selecionado pelo *ensemble* seria $\oplus$. Na composição majoritária ponderada, a classe selecionada seria $\ominus$.

De modo geral, a estratégia de voto majoritário com peso é definida da seguinte forma.

Definição 6.2: *Seja $\{h_1, h_2, \dots, h_L\}$ um ensemble sobre um mesmo espaço de classes $\{C_1, C_2, \dots, C_k\}$. Ao classificar um exemplo x , os classificadores são associados, respectivamente, aos pesos $\{w_1, w_2, \dots, w_L\}$. Cada previsão $h_i(x) = C_{h_i}$ é um voto com peso w_i para a classe C_{h_i} . O total de votos para cada classe é dado por*

$$V(C_j) = \sum_{i=1}^L w_i \cdot \mathbb{I}[C_{h_i} = C_j], \quad (6.2)$$

no qual $\mathbb{I}[\cdot]$ é a função indicadora (pág. 84).

O ensemble classificará x com a classe associada à maior soma de votos. Em caso de empate, pode-se selecionar aleatoriamente dentre uma das classes com maior peso ou utilizar algum outro critério previamente estabelecido de desempate.

O voto majoritário pode ser um mecanismo estático ou dinâmico. Quando ele é estático, os pesos w_i são selecionados durante a construção do *ensemble* e permanecem os mesmos para todos os exemplos classificados. Quando ele é dinâmico, os pesos w_i são calculados sempre que um novo exemplo é apresentado. Observe que o voto majoritário simples é um caso particular de voto ponderado estático no qual os pesos são idênticos para todos os classificadores.

Existem diversas razões para se definir pesos aos classificadores base. Por exemplo, podemos ter conhecimento prévio de que algum classificador é tipicamente mais fiel à função-

-conceito e, portanto, considerar que seu voto deve ter um peso maior que outros classificadores do *ensemble*. Algumas estratégias de definição de pesos são as seguintes.

Acurácia estimada: podemos avaliar todos os classificadores nos exemplos de treinamento e estimar sua acurácia esperada através de reamostragem. As acurácias estimadas podem ser empregadas como peso, representando “conhecimento prévio” de que alguns classificadores são provavelmente mais fiéis à função-conceito. Esta é a única estratégia de pesos estáticos empregada neste trabalho.

Distância como peso: ao classificar uma nova instância x , o classificador 1-NN atribui a ela o rótulo do seu vizinho mais próximo de acordo com algum conceito de vizinhança. Em algumas aplicações, o valor da distância entre essa instância e o seu vizinho mais próximo, $d(x, x_{v_1})$, pode ser utilizado como um valor de escore. Quanto menor a distância entre a instância e seu vizinho mais próximo, maior a confiança que pode ser dada na classificação. Em um *ensemble* de representações, a instância a ser classificada é comparada em vários domínios de representação. Para cada classificador base h_i , existe um vizinho mais próximo $x_{v_1}^{(h_i)}$. Para cada classificador base, atribui-se um peso de acordo com Equação 6.3.

$$w_i = \frac{1}{d(x, x_{v_1}^{(h_i)})} \quad (6.3)$$

Um problema dessa abordagem é que as distâncias geralmente não são comparáveis em diferentes representações. Uma solução para esse problema é normalizar as distâncias pelo “tamanho” estimado do espaço de decisão. Sejam z_α e z_β dois exemplos de treinamento tais que a distância entre eles é máxima, de acordo com o domínio de representação e a função e distância empregada no classificador base. Isto é, z_α e z_β são duas instâncias dissonantes (*discords*) no conjunto de treinamento, não havendo outro par de exemplos de treinamento com distância superior. Nesse caso, o peso pode ser normalizado como na Equação 6.4.

$$w_i = \frac{d(z_\alpha, z_\beta)}{d(x, x_{v_1}^{(h_i)})} \quad (6.4)$$

Os pesos w_i obtidos por esse procedimento são valores de escore $w_i > 1$. Note que, se a distância entre o exemplo de interesse e o seu vizinho mais próximo em algum classificador base for $d(x, x_{v_1}^{(h_i)}) = 0$, então seu peso w_i de acordo com a Equação 6.4 será não numérico. Nesse caso, pode-se atribuir a esse classificador base um valor de peso igual ou superior (com um limiar pré-estabelecido) ao maior valor de peso numericamente válido dentre os demais classificadores. No caso (improvável) de que $d(x, x_{v_1}^{(h_i)}) = 0$ para todos os classificadores base, então essa instância é classificada de acordo com o voto majoritário simples.

Probabilidade a posteriori: refere-se à probabilidade a posteriori de que o exemplo pertença a uma classe, dada sua vizinhança. Seja C_{h_i} a classe atribuída ao exemplo de interesse x pelo

i -ésimo classificador base. Então a probabilidade a posteriori $P(C_{h_i}|\mathbf{x})$ de que o exemplo realmente pertença à classe rotulada pode ser utilizada como um estimador de confiança do classificador base.

Como explicado na Seção 3.4.3, o classificador k -NN está baseado na regra dos “ k vizinhos mais próximos”. Dado um espaço de dados X , existe alta probabilidade de que uma instância $\mathbf{x} \in X$ será encontrada em uma região de vizinhança ocupada por exemplos da mesma classe (COVER; HART, 1967). Dessa forma, a probabilidade a posteriori de uma instância pode ser utilizada como um indicador de “consistência” da vizinhança de $\mathbf{x}$.

Uma forma de estimar a probabilidade a posteriori de um classificador é utilizar as classes que aparecem na vizinhança do exemplo de interesse $\mathbf{x}$. Seja $V = \{C_{v_1}, C_{v_2}, \dots, C_{v_k}\}$ o conjunto das classes dos k vizinhos mais próximos de $\mathbf{x}$ no conjunto de treinamento, de acordo com um classificador base h_i . Então, o peso associado a esse classificador é dado por

$$w_i = \frac{|\{C_j \text{ tais que } C_{v_j} = C_{v_1}\}_{j=1}^k|}{k}. \quad (6.5)$$

O peso w_i atribuído ao classificador base h_i é um escore no intervalo $[\frac{1}{k}, 1]$. Esse esquema de votos favorece domínios de representação nos quais o exemplo de interesse seja posicionado e regiões de menor incerteza. Uma desvantagem dessa abordagem é que as fronteiras inter-classe podem ser muito extensas em domínios de aplicação com muitas classes, de modo que a maioria das instâncias será posicionada em regiões de incerteza. Além disso, o número de vizinhos a ser utilizado no cálculo da probabilidade a posteriori é um parâmetro que deve ser selecionado para cada domínio de aplicação.

Uma simples extensão dos métodos apresentados consiste em utilizar um ponto de corte. Os classificadores que estiverem abaixo do ponto de corte podem ser desconsiderados na classificação de uma instância específica (no caso de pesos dinâmicos) ou removidos do *ensemble* para um domínio de aplicação (no caso de pesos estáticos). Na prática, isso equivale definir um conjunto de L' classificadores base e selecionar um subconjunto de $L < L'$ classificadores base.

O ponto de corte pode ser rígido ou flexível. Um ponto de corte rígido é um valor k tal que os pesos inferiores ao k -ésimo maior peso são desconsiderados (recebem peso $w_i = 0$). Isto é, se os pesos forem tais que $w_1 \geq w_2 \geq w_3 \dots \geq w_{L'}$, então os menores pesos são eliminados assumindo-se os novos pesos $w'_k = w'_{k+1} = \dots = w'_{L'} = 0$.

Um ponto de corte flexível é um limiar $0 < \delta < 1$. Esse ponto de corte define uma “tolerância” com respeito ao maior peso. Suponha que w_1 é o maior peso associado aos classificadores durante a rotulação de uma nova instância. Então, todos os pesos w_i tais que $w_1 - w_i > \delta$ são desconsiderados, assumindo-se o novo peso $w'_i = 0$ em seu lugar.

A semântica envolvida em utilizar um ponto de corte depende da semântica da estratégia de atribuição de pesos. Por exemplo, caso os pesos sejam escolhidos de acordo com a acurácia

do classificador no conjunto de treinamento, então um ponto de corte rígido representa a decisão de utilizar apenas os modelos mais fiéis à função-conceito. Por outro lado, um ponto de corte flexível em uma estratégia de probabilidade a posteriori expressa a decisão de utilizar apenas os classificadores que forneçam maior confiança para a rotulação de um exemplo, observado um limiar de tolerância.

6.4.2 Composição de rankings

Um problema semelhante à classificação é o *label ranking*. Enquanto o objetivo do classificador é encontrar a classe verdadeira de uma instância x , o propósito do *label ranker* é estabelecer uma ordem parcial entre os rótulos mais provavelmente associados à instância (ZHOU *et al.*, 2014).

Em alguns contextos, esse tipo de *ranking* pode ser utilizado para indicar que a instância pertence a mais de uma classe simultaneamente, sendo mais pertinente às classes com menor *rank*. Alternativamente, o *ranking* pode indicar uma incerteza na classificação, sendo que as classes com menor *rank* são provavelmente as mais apropriadas para o exemplo de interesse.

Formalmente, define-se um *ranking* de classes como o seguinte.

Definição 6.3: Seja (X, Y) um espaço de decisão no qual X é um espaço de objetos e $Y = \{C_1, C_2, \dots, C_k\}$ é um espaço de classes. Dado um exemplo $x \in X$, um *ranking* sobre as classes de x é um conjunto ordenado $R = \{r_1, r_2, \dots, r_k\}$ tal que:

- cada r_i é um rank associado a uma classe C_i ;
- dados r_i e r_j , observa-se $r_i < r_j$ se x pertence à classe C_i com maior probabilidade que C_j ;
- a soma de todos os ranks é constante para qualquer ranking. Isto é:

$$\sum_{i=1}^k r_i = \sum_{i=1}^k i.$$

Note que os *ranks* não necessariamente assumem valores que formam a sequência $1, 2, \dots, k$. Por exemplo, se uma instância pertence às classes C_1, C_2 e C_3 com probabilidades 40%, 20% e 40%, respectivamente, então um *ranking* de classes associado a essa instância poderia ser $R = \{1,5; 3; 1,5\}$.

O classificador 1-NN não é, fundamentalmente, um classificador voltado para *label ranking*, mas ele pode ser facilmente adaptado para essa tarefa. Uma forma simples de se obter um *ranking* é remover todos os exemplos que pertencem à mesma classe que o vizinho mais próximo e, então, repetir o processo. Com isso obtém-se o “vizinho mais próximo na ausência da primeira classe”. Esse processo pode ser repetido iterativamente até que vizinhos de todas as

classes tenham sido encontrados, como mostrado no Algoritmo 6.1. Note que se uma instância possuir dois vizinhos mais próximos à mesma distância e eles pertencerem a classes distintas, então o algoritmo atribui um *rank* médio a essas classes.

Algoritmo 6.1: Variação k -NN para *rankings*.

```

1 função  $R = \text{knn\_rank}(D, d, x)$ 
  Parâmetros de Entrada:
     $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$ : um conjunto de treinamento
     $d$ : uma função de distância  $d: X \times X \mapsto \mathbb{R}$ 
     $x$ : um exemplo de interesse
  Parâmetros de Saída:
     $R$ : um vetor de  $k$  elementos
  Pós-condições:
    Cada elemento  $R[i]$  é um rank associado a uma classe  $y_i$ 
2 PrimeiroRank  $\leftarrow 1$ 
3 enquanto  $D \neq \emptyset$  faça
4   Classes  $\leftarrow \emptyset$ 
5   MinDist  $\leftarrow \infty$ 
6   para cada exemplo de treinamento  $(x_i, y_i) \in D$  faça
7     Dist  $\leftarrow d(x, x_i)$ 
8     se Dist < MinDist então
9       Classes  $\leftarrow \{y_i\}$ 
10      MinDist  $\leftarrow$  Dist
11     se Dist = MinDist então
12       Classes  $\leftarrow$  Classes  $\cup \{y_i\}$ 
13   fim
14 fim
15 ÚltimoRank  $\leftarrow$  PrimeiroRank + |Classes| - 1
16 Rank  $\leftarrow \frac{\text{PrimeiroRank} + \text{ÚltimoRank}}{2}$ 
17 Atribua  $R[i] = \text{Rank}$  para cada classe  $y_i \in \text{Classes}$ 
18 Remova de  $D$  todos os exemplos pertencentes às classes presentes em Classes
19 PrimeiroRank  $\leftarrow$  ÚltimoRank + 1
20 fim

```

Neste trabalho, *rankings* são utilizados como um passo intermediário para estabelecer *ensembles* de representações. Nesse tipo de *ensemble*, cada classificador base vota em um *ranking* das classes. A etapa de composição de votos é substituída por um procedimento de composição de *rankings*, no qual as classes são associadas às médias dos *rank*s obtidos em cada classificador base. O *ensemble* em si é um classificador, não um *label ranker* e, após a composição dos *rankings*, ele seleciona para a instância de interesse a classe com menor valor de *rank* médio.

Definição 6.4: Seja $\{h_1, h_2, \dots, h_L\}$ um conjunto de classificadores base que compõem um *ensemble* referente um espaço de decisão (X, Y) tal que $Y = \{C_1, C_2, \dots, C_k\}$. Sejam $R_1, R_2, \dots, R_L$ os *rankings* de classes atribuídos a um exemplo de interesse x pelos classificadores base, tais que $R_j = \{r_{j1}, r_{j2}, \dots, r_{jk}\}$ são os votos do j -ésimo classificador base. A composição de votos

realizada pelo ensemble é obtida através da média dos ranks, isto é:

$$\bar{R} = \{\bar{r}_1, \bar{r}_2, \dots, \bar{r}_k\}, \quad (6.6)$$

tal que

$$\bar{r}_i = \sum_{j=1}^L \frac{r_{ji}}{L} \quad (6.7)$$

é a média dos ranks atribuídos à classe C_i pelos classificadores base. Então a classe atribuída por esse ensemble à instância x será aquela associada ao menor rank $\bar{r}_i$. Em caso de empate, escolhe-se aleatoriamente uma dentre as classes empatadas com igual probabilidade.

Resta definir o procedimento empregado para utilizar os classificadores base na construção dos *rankings*. Três métodos foram estudados neste trabalho:

Ordenação: trata-se da estratégia de selecionar iterativamente vizinhos de alguma classe. Dada uma instância de interesse x , encontra-se o seu vizinho mais próximo x_{v_1} , cuja classe é C_{v_1} . Atribuí-se a essa classe o rank $r_{v_1} = 1$. Em seguida, desconsiderando-se todos os exemplos de treinamento que pertencem à classe C_{v_1} , encontra-se o vizinho x_{v_2} , cuja classe é C_{v_2} . A essa classe é atribuído o rank $r_{v_2} = 2$. Esse procedimento é realizado iterativamente, encontrando-se os ranks $r_{v_j} = j$ até que todas as classes tenham sido associadas a um rank.

Probabilidade a posteriori: a probabilidade a posteriori de que uma instância de interesse pertença a cada classe pode ser utilizada para estabelecer uma ordem parcial e um *ranking* de classes. Um método para estimar probabilidades a posteriori utilizando k -NN foi proposto por [Atiya \(2005\)](#), no qual os vizinhos de uma instância são associados a pesos $\{s_1, s_2, \dots, s_k\}$ tais que $s_i \in [0, 1]$ e $\sum s_i = 1$. Cada s_i expressa a significância do i -ésimo vizinho mais próximo de x na estimativa da sua probabilidade a posteriori. A probabilidade de que uma instância x pertença à classe C_i é estimada como a soma ponderada dos pesos associados aos vizinhos de x que pertencem a essa classe. Considere, por exemplo, um caso no qual as probabilidades são estimadas de uma vizinhança com $k = 5$ e um espaço de classes $Y = \{C_1, C_2, C_3, C_4\}$. Suponha que cinco vizinhos mais próximos de x pertencem, ordenadamente, às classes C_1, C_2, C_1, C_4 e C_2 . Assim, as probabilidades de que essa instância pertença a cada classe são:

Classe	Probabilidade
C_1	$s_1 + s_3$
C_2	$s_2 + s_5$
C_3	0
C_4	s_4

As probabilidades a posteriori obtidas para as classes $P(C_i|x)$ são então ordenadas e um *ranking* é estabelecido. Note que não existem, dentre os cinco vizinhos mais próximos de x ,

exemplos da classe C_3 . De modo geral, quanto maior o número de classes associadas a um domínio de aplicação, maior deve ser o tamanho da vizinhança empregada na estimativa das probabilidades a posteriori.

Para definir os pesos, [Atiya \(2005\)](#) propõe um modelo de otimização baseado na máxima verossimilhança dos exemplos de treinamento. O modelo é iniciado com um conjunto de pesos homogêneo e, a cada iteração, é considerada a probabilidade de que as vizinhanças observadas sejam compatíveis com os pesos. Os pesos são ajustados de acordo com o gradiente da função de verossimilhança até que a diferença de pesos entre duas iterações seja inferior a um limiar ou até que um número máximo de iterações seja realizado.

Composição baseada em *simple-ranking*: o *simple-ranking* é uma estratégia de classificação *anytime* que estabelece uma ordem de prioridade para os exemplos de treinamento ([UENO et al., 2006](#)). Em classificação *anytime*, existem restrições ao tempo de processamento que pode ser dedicado à classificação de uma nova instância. O *simple-ranking* ordena os exemplos de treinamento de modo que uma instância de interesse x é comparada primeiramente com os exemplos que mais provavelmente se encontram na vizinhança de x . Desse modo, se o classificador for interrompido antes que toda a vizinhança possa ter sido explorada, a probabilidade de que o vizinho mais próximo (ou alguma outra instância semelhante ao vizinho mais próximo) tenha sido encontrado durante o tempo de processamento é maior do que uma implementação baseada em *linear scan*.

O procedimento utilizado para produzir um *simple-ranking* possui paralelos com uma validação cruzada *leave-one-out*. Seja $D = \{S_1, S_2, \dots, S_N\}$ um conjunto de séries temporais. Inicialmente, o *simple-ranking* atribui um escore $f_i = 0$ para cada exemplo de treinamento. Em seguida, o exemplo de treinamento $s[1]$ é temporariamente removido do conjunto e as demais séries são utilizadas para classificá-lo. O vizinho mais próximo desse exemplo é denominado um “aliado” se pertencer à mesma classe de $s[1]$ —isto é, se esse vizinho contribuir corretamente para a classificação desse exemplo. Caso contrário, ele é considerado um “adversário”. O exemplo é devolvido ao conjunto de treinamento e o procedimento é realizado para os próximos exemplos $s[2], s[3], \dots, s[N]$. Sempre que um vizinho $s[v_1]$ é considerado um “aliado”, seu escore é incrementado em 1. Quando um vizinho é considerado um “adversário”, seu escore é decrementado em $\frac{2}{k-1}$, no qual k é o número de classes no espaço de decisão. Ao final desse procedimento, os exemplos de treinamento são ordenados de acordo com os escores e um *ranking* é estabelecido.

Ao classificar um exemplo de interesse x , os *ranks* associados aos exemplos de treinamento são utilizados de forma semelhante aos pesos no *ensemble* baseado em probabilidade a posteriori. Sejam $s_1, s_2, \dots, s_k$ os *ranks* associados aos k vizinhos mais próximos de x . Para cada classe C_j , um novo escore é calculado como a média dos *ranks* dos vizinhos que possuem classe C_j . Considere novamente o caso no qual os cinco vizinhos mais próximos

pertencem às classes C_1 , C_2 , C_1 , C_4 e C_2 . Então, os escores atribuídos às classes são os seguintes.

Classe	Score
C_1	$S_1 = \frac{s_1 + s_3}{2}$
C_2	$S_2 = \frac{s_2 + s_5}{2}$
C_3	$S_3 = 0$
C_4	$S_4 = s_4$

Com base nos escores atribuídos às classes, um novo *ranking* é estabelecido, favorecendo escores inferiores. Esse *ranking* é utilizado como o voto do classificador base.

Note que os *ranks* médios obtidos por esse procedimento estabelecem uma ordem parcial entre as classes e poderiam ser utilizados para realizar *label ranking*. Entretanto, mantendo o escopo desta Tese, o procedimento de composição de *rankings* é utilizado apenas como etapa intermediária da classificação por *ensembles*.

6.5 Avaliação experimental

Durante a execução deste trabalho, foram estudadas diversas representações de séries temporais. No Capítulo 5, os experimentos apresentados tiveram como objetivo procurar representações que permitissem construir classificadores 1-NN com maior acurácia do que o classificador 1-NN DTW ou, pelo menos, que permitissem prever, adequadamente, quando uma representação específica permitiria produzir um classificador superior.

Neste capítulo, apresenta-se experimentos nos quais o objetivo é compor representações com o objetivo de construir classificadores de maior acurácia. São apresentados experimentos que foram conduzidos em duas fases diferentes da realização deste trabalho. O primeiro conjunto de experimentos, apresentado na Seção 6.5.1, foi realizado sobre os 45 conjuntos do repositório UCR que estavam disponíveis à época. Os experimentos seguintes foram realizados após a atualização do repositório e neles são utilizados um maior número de conjuntos e de representações.

6.5.1 Comparação de mecanismos de voto

O objetivo almejado na utilização de *ensembles*, neste trabalho, era explorar, simultaneamente, diferentes domínios de representação de dados temporais como mecanismos de expor diferentes atributos discriminativos das classes. Por isso, todos os classificadores base são variações do 1-NN sobre diferentes representações dos mesmos dados. Os resultados experimentais apresentados nesta seção foram publicados em (GIUSTI; SILVA; BATISTA, 2015).

Os experimentos foram realizados nos 45 conjuntos de dados do repositório UCR que existiam à época. Foram consideradas quatro representações e, no domínio temporal, foram

consideradas a distância euclidiana e a distância DTW, resultando nos seguintes classificadores base:

- 1-NN;
- 1-NN DTW sem janela;
- 1-NN DTW com janela Sakoe-Chiba;
- 1-NN com representação FS;
- 1-NN com representação ACF;
- 1-NN com *wavelets* de Haar.

Note que apenas a *wavelet* de Haar foi utilizada nesse experimento, não sendo verificadas outras bases da família de Daubechies. As outras bases foram avaliadas em experimentos posteriores.

Os experimentos descritos nesta Tese foram realizados com os seis classificadores base. Todos os métodos de composição de votos apresentados na Seção 6.4 foram avaliados. Foram utilizados três vizinhos nos dois *ensembles* baseados em probabilidade a posteriori e no *simple rank*. Esses *ensembles* foram comparados entre si e também com o classificador 1-NN DTW através de um teste de hipótese de Friedman, seguido de um teste de Nemenyi. Os resultados completos são apresentados no apêndice (Seção A.8, pág. 231). Na Figura 60 mostra-se um gráfico CD resultante do teste de hipótese.

Figura 60 – Gráfico de distância crítica para todos os métodos de composição de votos.

Os resultados indicam grande variância entre os diferentes métodos de composição de votos e também que a maioria desses métodos consegue produzir classificadores capazes de superar o 1-NN DTW no caso médio. Esse não é um resultado inesperado, pois o próprio 1-NN DTW está presente entre os classificadores base do *ensemble*, mas é indicativo de que incluir diversidade de representações pode melhorar o desempenho do classificador 1-NN no domínio do tempo.

Com base nesses resultados, foi conduzido um experimento de seleção de classificador. Todos os métodos de composição de votos foram avaliados no conjunto de treinamento utilizando

validação 5×2 CV. Para os métodos baseados em *simple rank* e probabilidade a posteriori, os experimentos foram realizados com $k = 3$. Para cada conjunto de dados, o *ensemble* dinâmico seleciona a melhor estratégia de voto no conjunto de treinamento e a utiliza no conjunto de teste. Os resultados obtidos são apresentados no apêndice (Seção A.7, pág. 226). Na Figura 61, são apresentados gráficos de espalhamento comparando o *ensemble* dinâmico, o 1-NN DTW e o *ensemble* majoritário ponderado pela acurácia.

Figura 61 – Acurácia do *ensemble* dinâmico.

Na Figura 61-direita, observa-se que o *ensemble* dinâmico possui desempenho muito próximo, mas consistentemente superior ao 1-NN DTW. Por outro lado, quando se compara o *ensemble* dinâmico com o *ensemble* majoritário ponderado pela acurácia (doravante simplesmente “*ensemble* de acurácias”), percebe-se que o ganho obtido pelo método de seleção é pequeno (Figura 61-esquerda). A interpretação visual é confirmada pelo teste de hipótese de Wilcoxon. Por esse motivo, optou-se por utilizar o *ensemble* de acurácias em detrimento dos demais métodos de composição de votos.

6.5.2 “Atualização” dos experimentos

Posteriormente aos experimentos relatados na seção anterior, foram acrescentadas novas representações de séries temporais e novas funções de distância. O repositório UCR foi atualizado com 40 novos conjuntos de dados e renormalização de alguns conjuntos (Seção 2.6.1). Portanto, os experimentos de *ensembles* foram repetidos conforme apropriado.

No primeiro conjunto de experimentos, foram mantidas as representações temporal, FS e ACF. Porém, a representação ACF foi tomada com os parâmetros ajustados de acordo com experimentos relatados na Seção 5.8.1. Para o domínio tempo-frequência, a representação de Haar foi substituída pela DWT e uma base adequada foi escolhida para cada conjunto. Observe que, no domínio temporal, ainda foram utilizadas as mesmas funções de distância: euclidiana, DTW sem janela e DTW com janela Sakoe-Chiba. Uma comparação desse *ensemble* com o 1-NN DTW é apresentada na Figura 62-esquerda.

Ainda considerando as quatro representações do conjunto anterior, promoveu-se seleção de função de distância. As funções de distância mais adequadas para cada representação foram selecionadas de acordo com experimentos apresentados na Seção 5.8.2. Então, neste experimento, os classificadores com distância euclidiana foram “atualizados” conforme necessário. Uma comparação do *ensemble* resultante com o *ensemble* de acurácias anterior é feita na Figura 62-direita.

Figura 62 – *Ensemble* de acurácias e seleção de distâncias. À esquerda, compara-se o *ensemble* de acurácias contra o 1-NN DTW nos 85 conjuntos do repositório UCR. À direita, duas versões do *ensemble* de acurácias: uma na qual a melhor função de distância foi escolhida e outra na qual foram utilizadas apenas as distâncias euclidiana e DTW.

Em seguida, foram avaliados *ensembles* utilizando todas as representações consideradas neste projeto. Aquelas que possuem parâmetros foram previamente ajustadas de acordo com os experimentos relatados na Seção 5.8.1. Portanto, além dos seis classificadores base listados na seção anterior, foram adicionados o 1-NN com distância euclidiana nas representações PAA, PCA e SAX. Essa configuração foi denominada “*ensemble completo*”. Como se pode observar na Figura 63, embora o *ensemble completo* tenha *acrescentado* representações, os resultados foram *inferiores* ao *ensemble* de acurácias.

Levantou-se então a hipótese de que selecionar distâncias mais apropriadas para as representações alternativas poderia aumentar a acurácia do *ensemble completo*. O mesmo procedimento de seleção aplicado ao *ensemble* de acurácias foi repetido para o *ensemble completo*. Os resultados indicaram que o acréscimo das representações não se traduziu em maior qualidade de classificação, como mostrado na Figura 64.

Foram conduzidos experimentos preliminares visando explorar outros meios de aumentar a acurácia do *ensemble* de representações. Porém, nenhum resultado superior ao *ensemble* de acurácias com seleção de distâncias foi obtido no caso médio. Como esses experimentos preliminares envolvem representações que não foram originalmente incluídas e discutidas nos experimentos do Capítulo 5, essa discussão será realizada na Seção 7.2, “Contribuições e

Figura 63 – *Ensemble* completo contra *ensemble* de acurácias.

Figura 64 – *Ensemble* completo, com seleção de distâncias, contra *ensemble* de acurácias, sem seleção de distâncias.

limitações”.

6.6 Considerações finais

A composição de modelos de classificação em comitês ou *ensembles* de classificadores é uma estratégia frequentemente encontrada no aprendizado de máquina. Especialmente em problemas “difíceis”, a mistura de diferentes tipos de vies frequentemente permite a obtenção de resultados mais precisos do que quando se utiliza um único classificador.

Nesta Tese, foram consideradas seis técnicas de construção de *ensembles* e oito domínios de representação temporal, aos quais foram incorporadas 17 funções de distância. A hipótese original era que as técnicas mais sofisticadas, em particular as de composição de *rankings*, seriam

capazes de melhor explorar o espaço de decisão e encontrar o classificador de maior acurácia. Os resultados obtidos, no entanto, foram inesperados, sugerindo que estratégias mais simples podem ser suficientes para se obter classificadores bastante acurados.

Os métodos de construção de *ensemble* utilizados nos experimentos desta Tese foram escolhidos com o objetivo principal de maximizar a diversidade de representações estudadas. Por questões de escopo e tempo, não foram utilizadas algumas técnicas de *ensemble* comuns, como *bagging* e *boosting*. O uso de representações distintas é dificultado quando os classificadores base não estão construídos sobre os mesmos exemplos (*e.g.*, no caso de reamostragem). Entretanto, é possível conceber um *ensemble* hierárquico, no qual cada classificador base é um *ensemble* sobre uma única representação utilizando *bagging* ou *boosting*, por exemplo. Esse é um ponto de pesquisa futuro.

Pelo conhecimento do autor, a consideração de simultânea de diversidade de representações e de experimentos é uma contribuição original. Embora outros autores tenham considerado funções de distância específicas para representações específicas (DING *et al.*, 2010), não há trabalhos nos quais um modelo de classificação do tipo “1-NN dinâmico” (como apresentado na Seção 4.7.2.2) tenha sido associado à diversidade de representações. O *ensemble* de acurácias com seleção de distâncias, obtido dessa associação, foi capaz de produzir resultados bastante acurados, superando o classificador 1-NN DTW e também o *ensemble* majoritário ponderado pela acurácia. Trata-se de um resultado bastante positivo, que sugere métodos de construção de *ensembles* com boa precisão e custo computacional muito mais baixo do que o estado-da-arte (BAGNALL *et al.*, 2015).

Com esta seção, encerram-se as discussões das atividades realizadas durante a execução deste projeto. No próximo capítulo é feita uma revisão e uma releitura da Tese.

CONCLUSÕES

7.1 Resumo do projeto e revisão do trabalho

A PRINCIPAL PROPOSTA DESTE TRABALHO é estudar a utilização de diferentes representações de dados na classificação de séries temporais. A motivação para essa pesquisa se sustenta na hipótese de que

“Não existe uma única representação de séries temporais capaz de evidenciar as características discriminativas das classes para todo e qualquer domínio de aplicação.”

Na primeira década deste século, diversos autores sustentaram a noção de que o classificador 1-NN DTW seria, no caso médio, o melhor algoritmo para classificação de dados temporais. Normalmente, trabalhos nos quais se considerava uma representação de dados alternativa eram limitados a estratégias para melhorar a eficiência do classificador (DING *et al.*, 2008). A representação alternativa deveria ser empregada com o objetivo de reduzir o espaço de busca, porém a decisão da classe deveria ocorrer no domínio temporal. A classificação em domínios alternativos era tipicamente restrita a aplicações envolvendo análise de sinais digitais (CHAOVALIT *et al.*, 2011), nos quais se utiliza coeficientes de frequência ou tempo-frequência.

Nos anos recentes, outros autores têm dado atenção para a possibilidade de se utilizar representações alternativas como espaço de decisão para o problema de classificação. Entretanto, grande parcela desses trabalhos foca em questões pontuais. Oates *et al.* (2012) é provavelmente o caso mais antigo no qual uma representação é utilizada para promover um espaço alternativo de classificação. No entanto, nesse trabalho apenas uma representação (SAX) foi considerada como mecanismo para extração de atributos temporais (*Bag of Patterns*). Outro trabalho mais recente é o *bag of SFA symbols* (BOSS), no qual são extraídos atributos da representação DFT. O BOSS é um classificador que obtém resultados excelentes no caso médio (SCHÄFER, 2015),

mas não fecha a lacuna existente na literatura com respeito à forma mais adequada de se utilizar diversidade de representações.

O trabalho mais semelhante a esta Tese é o “coletivo de *ensembles* de transformação” ou COTE (LINES; HILLS; BAGNALL, 2014). Trata-se de um *ensemble* hierárquico que combina diferentes estratégias de classificação utilizando quatro representações de séries temporais. Os autores também apresentam um modelo denominado “*flat-COTE*”, que é uma versão não hierárquica dos mesmos classificadores base empregados no COTE. Entretanto, uma limitação desse trabalho é o grande número de classificadores base utilizados, resultando em um modelo bastante complexo e computacionalmente muito mais custoso do que a proposta deste trabalho.

A principal diferença deste trabalho àquele apresentado em (LINES; HILLS; BAGNALL, 2014) é o foco em representações como espaços de decisão alternativos. Ao limitar o classificador a um *ensemble* homogêneo de classificadores 1-NN, nesta Tese efetivamente explora-se a diversidade de representações como mecanismo para expor as características discriminativas do domínio, sem influência de outros modelos de classificação. Os resultados obtidos demonstram que um classificador é substancialmente melhor quando emprega diferentes representações simultaneamente.

Os resultados desejados desta pesquisa podem ser sumarizados nos seguintes pontos:

- A obtenção de um classificador baseado em diversidade de representações com alto poder discriminativo;
- A obtenção de um *ensemble* de classificadores com alto poder discriminativo;
- O uso de diversidade de representações para construção de classificadores com alto poder discriminativo.

O autor acredita que, nesta Tese, apresenta-se um relato que evidencia que esses pontos foram alcançados.

No Capítulo 4, as funções de distância são utilizadas como mecanismo de se explorar diferentes espaços de decisão no domínio temporal. Foi mostrado como o uso de funções pouco convencionais, como a distância de Bhattacharyya, podem auxiliar na construção de classificadores com alta acurácia em muitos domínios de aplicação.

No Capítulo 5, foram considerados domínios de representação alternativos. Esse é um tópico que vem recebendo, recentemente, atenção considerável da comunidade de AM. Entretanto, o autor acredita, com base na revisão bibliográfica realizada ao longo deste trabalho, que o foco ainda é restrito a algumas poucas representações baseadas em frequência, tempo-frequência ou atributos relacionados (ACF). Ainda, este é o primeiro trabalho no qual a diversidade de representações foi utilizada para extrair coeficientes que podem ser utilizados no treinamento de qualquer modelo de classificação.

No Capítulo 6, foram exploradas diferentes estratégias de composição de classificadores em *ensembles* que utilizam diversidade de representações e distâncias na classificação de séries temporais. Embora este não seja o primeiro trabalho na literatura a utilizar diversidade de representações em *ensembles*, é provavelmente o primeiro no qual múltiplas técnicas de *ensemble* utilizando diversidade foram comparadas.

Como resultado direto desta pesquisa, foram publicados três artigos completos em conferências e, em trabalho relacionado, o autor contribuiu como co-autor de um trabalho publicado em conferência:

- Giusti, R.; Silva, D. F.; Batista, G. E. A. P. A. “Improved Time Series Classification with Representation Diversity and SVM”. In: 15th IEEE International Conference on Machine Learning and Applications, 2016, Anaheim, CA, EUA. Proceedings of the 15th IEEE International Conference on Machine Learning and Applications, 2016. pp. 1–6.
- Giusti, R.; Silva, D. F.; Batista, G. E. A. P. A. “Time Series Classification with Representation Ensembles”. In: Fourteenth International Symposium on Intelligent Data Analysis, 2015, Saint-Etienne. Advances in Intelligent Data Analysis XIV, 2015. pp. 108–119.
- Giusti, R.; Batista, G. E. A. P. A. “An Empirical Comparison of Dissimilarity Measures for Time Series Classification”. In 2nd Brazilian International Conference on Intelligent Systems (BRACIS). 2013, Fortaleza CE. Proceedings of the Brazilian Conference on Intelligent Systems, 2013. pp 82–88.
- Silva, D. F.; Souza, V. M. A.; Batista, G. E. A. P. A.; Giusti, R. “Spoken Digit Recognition in Portuguese Using Line Spectral Frequencies”. In: 13th Ibero-American Conference on Artificial Intelligence, 2012, Cartagena de Indias. Proceedings of the 13th Ibero-American Conference on Artificial Intelligence, 2012. v. 7637. pp. 241–250.

7.2 Principais contribuições e limitações

A principal proposta deste trabalho é o estudo de representações e distâncias que são normalmente desconsideradas na literatura de classificação de séries temporais. Nesta Tese, todas as representações e funções de distância são tratadas como espaços de busca. As principais contribuições originadas dessa proposta e dos trabalhos relacionados que foram identificados ao longo da pesquisa são os seguintes:

1. Foi conduzido um estudo das principais formas de representação de séries temporais. Das consideradas ao longo do trabalho, foram incluídas nesta Tese as mais importantes dos seguintes grupos: baseadas em frequência (DFT e FS), em tempo-frequência (DWT), segmentação (PAA), simbólica (SAX), rotação (PCA) e autocorrelação (ACF). No Capítulo 5

apresenta-se uma visão geral dessas representações e como elas podem ser utilizadas para construir espaços de decisão alternativos para classificação de séries temporais;

2. Foi feito um levantamento de funções de distâncias para séries temporais que podem ser utilizadas sobre as representações selecionadas. Os métodos e resultados apresentados nesta Tese são relevantes para a comunidade científica porque, no conhecimento do autor, não existem trabalhos anteriores que considerem simultaneamente o emprego de diversidade de representações e de funções de distância. Como foi mostrado nos Capítulos 5 e 4, essa associação permite a construção de classificadores bastante precisos e competitivos com o estado da arte em muitos casos. Muitas dessas distâncias são não elásticas, o que cria abertura para a estudos visando construir classificadores com alta acurácia e baixa complexidade computacional;
3. Foi realizado um estudo sobre o emprego de diferentes estratégias de normalização. No conhecimento do autor, a normalização em escores é simplesmente *assumida* pela comunidade de aprendizado de máquina como sendo a mais adequada para classificação de séries temporais. Os experimentos dessa Tese confirmam essa hipótese para o domínio temporal e a distância euclidiana e, principalmente, mostram que existem métodos alternativos de normalização que são muito mais adequados para algumas funções de distância pouco convencionais. Essa verificação é particularmente relevante para o item 2 desta lista;
4. Um método para extração de atributos não temporais que incorporam o conceito de vizinhança promovido por diversidade de representações e de distância. Esse resultado é animador para a pesquisa em classificação de séries temporais, pois normalmente tais conceitos são explorados unicamente com modelos especificamente baseados em vizinhança, como o 1-NN. Nesta Tese, foi mostrado como representações alternativas podem ser utilizadas para induzir modelos de classificação utilizando linguagens de descrição arbitrárias;
5. Uma proposta para a construção de *ensembles* de representações e, principalmente, um conjunto de recomendações para a classificação utilizando esse tipo de modelo composto. Foi observado que *ensembles* de representações são tipicamente mais eficazes quando utilizam um pequeno número de representações de diferentes grupos, em detrimento de *ensembles* utilizando muitas representações relacionadas;
6. Uma comparação do desempenho de diferentes métodos de construção de *ensembles* de classificadores, revelando que o *ensemble* majoritário ponderado é normalmente muito mais eficaz do que métodos mais complexos, como os baseados em composição de *rankings*.

De modo geral, todos os resultados apresentados neste trabalho podem ser considerados uma contribuição desta pesquisa. Os experimentos foram realizados cautelosamente e cobrem

escopos bastante extensos da pesquisa em classificação de séries temporais. São resultados que indicam tendências do classificador 1-NN utilizando as representações e distâncias selecionadas, conduzidos com o auxílio de vários computadores de alto desempenho, não disponíveis para muitos pesquisadores e profissionais em geral da área de mineração de dados. Todos esses resultados são publicamente disponibilizados em um *website*, juntamente com o *software* que foi desenvolvido, ao longo desta pesquisa, para a condução desses experimentos (GIUSTI, 2017).

Dentre as principais limitações encontradas durante a execução desta pesquisa, a mais importante diz respeito à escolha do repositório UCR para a condução dos experimentos. Embora, durante a execução deste projeto, outros conjuntos tenham sido adotados, a grande quantidade de dados desse repositório, sua prevalência nos trabalhos relacionados encontrados durante a revisão bibliográfica e a sua diversidade levaram à escolha desses conjuntos nos testes conduzidos. Porém, ao longo da pesquisa, foram observadas algumas deficiências nesse repositório. A mais relevante é que todos os conjuntos são normalizados e pré-processados para que todas as séries possuam o mesmo número de observações. Como já foi apontado por vários pesquisadores, tal cenário de classificação dificilmente ocorre em uma aplicação real. Além disso, o fato de que esses conjuntos são previamente distribuídos com os dados normalizados em escores padrões reduz o escopo de pesquisa em técnicas de pré-processamento dos dados.

Outra limitação foi a utilização exclusiva de técnicas de processamento em memória principal. Um dos objetivos originais deste trabalho era o estudo de técnicas de indexação e, de modo geral, classificação em memória secundária. A necessidade de se acessar dados em memória secundária é um grande desafio na computação, principalmente quando esse acesso não pode ser feito de forma linear. Entretanto, o esforço dedicado nesta pesquisa se concentrou principalmente no estudo das representações, de modo que esse tema de pesquisa não foi executado.

Muito tempo foi investido em tópicos de pesquisa que não resultaram nos resultados esperados. Por exemplo, foi realizado um longo estudo sobre os coeficientes da representação PCA como uma forma simples de se obter um classificador baseado em projeções. A representação PCA é obtida quando se encontra uma matriz de mudança de base sobre todos os exemplos de treinamento. O que poderia ser feito se fossem obtidas diversas matrizes de mudança de base para as várias classes do domínio? A hipótese levantada era que, se um exemplo pertencente a uma classe C_i fosse rotacionado de acordo com uma matriz otimizada para os exemplos de treinamento da classe C_j , a série resultante seria uma dissonante. Assim, foram pesquisados diferentes métodos nos quais um exemplo de classe desconhecida seria rotacionado em espaços determinados por diferentes bases. Mas os classificadores obtidos não apresentaram a acurácia esperada e, como se trata de uma aplicação muito específica a uma única representação, a pesquisa foi interrompida.

Na Seção 6.5.2, comentou-se que foi feita uma tentativa de incluir representações ao *ensemble* de acurácias com representações que não foram incluídas no trabalho final. Essas

representações foram a transformada discreta de cossenos (AHMED; NATARAJAN; RAO, 1974), a representação *bag of patterns* (LIN; KHADE; LI, 2012), a representação *time series bitmaps* (KUMAR *et al.*, 2005) e a transformação de *shapelets* (YE; KEOGH, 2011). A transformada discreta de cossenos foi abandonada porque não foi capaz de melhorar a acurácia do *ensemble* e por ser uma transformação baseada em frequências, domínio de representação que já estava sendo explorado com a transformada discreta de Fourier. A *bag of patterns*, apesar de ser uma representação com potencial para produzir classificadores precisos, foi excluída das etapas posteriores da pesquisa devido ao alto custo de tempo e espaço envolvido na construção e no uso de classificadores baseados nessa representação. A representação *time series bitmaps* é muito similar semelhantes à *bag of patterns*, mas não produz resultados que se justifiquem seu custo computacional. A representação baseada em *shapelets*, também a despeito de sua acurácia, não foi utilizada devido ao elevado custo computacional exigido para encontrar os *shapelets*. Obviamente, existem outros trabalhos nos quais essas representações *são* utilizadas. Escolheu-se não empregá-las para que se pudesse dedicar mais esforços às representações apresentadas no Capítulo 5. Não obstante, mesmo que justificável, essa é uma limitação deste trabalho que não pode ser ignorada.

7.3 Possibilidades de trabalhos futuros

Uma extensão direta desta pesquisa é a utilização de novos domínios de representação e funções de distância. Na seção anterior, foi comentado que algumas representações foram abandonadas devido ao alto custo computacional. Entretanto, devido a restrições de tempo, não foram pesquisadas *aproximações* dessas representações. Além disso, existem diversas outras representações, algumas bastante semelhantes às empregadas neste trabalho, que podem ser utilizadas em *ensembles* de representações. O autor pretende prosseguir o estudo de diversidade de representações estendendo a configuração de *ensembles* apresentada no Capítulo 6.

Outra forma de estender o *ensemble* de representações será o emprego de métodos de reamostragem, como como *bagging* e *boosting* (FLACH, 2012). Uma provável maneira de se atingir esse fim é com o uso de estruturas hierárquicas de *ensembles*. O autor pretende realizar tal estudo, especialmente no uso de diferentes configurações de uma mesma representação.

Finalmente, existe a questão do procedimento de ajuste de parâmetros. Ao longo desta Tese, foram mostrados vários casos de ajuste de parâmetros por validação cruzada *leave-one-out*. O problema encontrado durante a realização desses experimentos é o conhecido compromisso existente entre viés e variância (RUSSELL; NORVIG, 2010). Teoricamente, é impossível conceber um classificador que acerte todos os exemplos de teste para domínios arbitrários de aplicação. Consequentemente, é teoricamente impossível conceber um método de seleção de parâmetros ou algoritmos que sempre seja capaz de encontrar a configuração ideal para domínios arbitrários de aplicação. Entretanto, é teoricamente possível obter resultados *melhores*. Em (BATISTA

et al., 2013), foi apresentado o conceito de gráfico *sharpshoot*: uma contagem de casos de verdadeiros-positivos, falsos-positivos, verdadeiros-negativos e falsos-negativos encontrados durante os estágios de ajuste e avaliação de configurações de um algoritmo. Esse gráfico foi proposto precisamente com o objetivo de fornecer uma intuição visual de algoritmos *previsíveis*. Um ponto de pesquisa futuro é utilizar procedimentos de reamostragem e contagem de casos verdadeiro/falso-positivo/negativo como forma de se encontrar representações e funções de distância *estáveis* para um domínio de aplicação. A hipótese é que, dado um grande número de representações e distâncias candidatas, essa contagem permita escolher um subconjunto de representações e distâncias nas quais o procedimento de ajuste de parâmetros permita encontrar a configuração ótima com maior probabilidade. Isso certamente não *eliminará* o problema da variância, mas supõe-se que permitira obter-se classificadores mais acurados. Esse é um tópico de pesquisa que foi considerado durante a execução deste trabalho. Experimentos preliminares foram conduzidos, mas optou-se por concentrar nas demais atividades concluídas e apresentadas nesta Tese.

REFERÊNCIAS

- AGHABOZORGI, S.; SHIRKHORSHIDI, A. S.; WAH, T. Y. Time-series clustering—a decade review. **Information Systems**, Elsevier, v. 53, p. 16–38, 2015. Citado na página 52.
- AGRAWAL, R.; FALOUTSOS, C.; SWAMI, A. Efficient similarity search in sequence databases. **Foundations of Data Organization and Algorithms**, Springer, p. 69–84, 1993. Citado nas páginas 25, 26 e 141.
- AHMED, N.; NATARAJAN, T.; RAO, K. Discrete cosine transform. **Computers, IEEE Transactions on**, C-23, n. 1, p. 90–93, Jan 1974. ISSN 0018-9340. Citado na página 196.
- AL-JOWDER, O.; KEMSLEY, E.; WILSON, R. H. Detection of adulteration in cooked meat products by mid-infrared spectroscopy. **Journal of agricultural and food chemistry**, ACS Publications, v. 50, n. 6, p. 1325–1329, 2002. Citado na página 50.
- ALPAYDIN, E. **Introduction to Machine Learning**. 2. ed. [S.l.]: Massachusetts Institute of Technology, 2010. Citado nas páginas 63, 84 e 172.
- ANDREÃO, R. V.; DORIZZI, B.; BOUDY, J. Ecg signal analysis through hidden markov models. **IEEE Transactions on Biomedical engineering**, IEEE, v. 53, n. 8, p. 1541–1549, 2006. Citado na página 49.
- ANGUITA, D.; GHIO, A.; ONETO, L.; PARRA, X.; REYES-ORTIZ, J. L. A public domain dataset for human activity recognition using smartphones. In: **ESANN**. [S.l.: s.n.], 2013. Citado na página 60.
- ASSENT, I.; WICHTERICH, M.; KRIEGER, R.; KREMER, H.; SEIDL, T. Anticipatory dtw for efficient similarity search in time series databases. **Proceedings of the VLDB Endowment**, VLDB Endowment, v. 2, n. 1, p. 826–837, 2009. Citado na página 122.
- ATIYA, A. F. Estimating the posterior probabilities using the k-nearest neighbor rule. **Neural Computation**, MIT Press, v. 17, n. 3, p. 731–740, mar. 2005. ISSN 0899-7667. Disponível em: <<http://dx.doi.org/10.1162/0899766053019971>>. Citado nas páginas 182 e 183.
- BACHMAN, G.; NARICI, L.; BECKENSTEIN, E. **Fourier and Wavelet Analysis**. [S.l.]: Spr, 2000. Citado nas páginas 138, 145 e 146.
- BAGNALL, A.; DAVIS, L.; HILLS, J.; LINES, J. Transformation based ensembles for time series classification. **Proceedings of the 12th SIAM International Conference on Data Mining**, p. 307–319, 2012. Citado nas páginas 50 e 90.
- BAGNALL, A.; LINES, J. An experimental evaluation of nearest neighbour time series classification. **arXiv preprint arXiv:1406.4757**, 2014. Citado na página 80.
- BAGNALL, A.; LINES, J.; HILLS, J.; BOSTROM, A. Time-series classification with cote: the collective of transformation-based ensembles. **IEEE Transactions on Knowledge and Data Engineering**, IEEE, v. 27, n. 9, p. 2522–2535, 2015. Citado nas páginas 101, 105, 158, 176 e 189.

- BALLOU, G. **Handbook for sound engineers**. [S.l.]: Taylor & Francis, 2013. Citado na página 41.
- BATISTA, G.; WANG, X.; KEOGH, E. A complexity-invariant distance measure for time series. In: **SIAM Conf. Data Mining**. [S.l.: s.n.], 2011. p. 699–710. Citado nas páginas 41, 42, 43, 45, 46, 92, 130 e 158.
- BATISTA, G. E. A. P. A.; KEOGH, E. J.; MAFRA-NETO, A.; ROWTON, E. Sensors and software to allow computational entomology, an emerging application of data mining. In: APTÉ, C.; GHOSH, J.; SMYTH, P. (Ed.). **Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, August 21-24, 2011**. [S.l.]: ACM, 2011. p. 761–764. ISBN 978-1-4503-0813-7. Citado nas páginas 26, 27, 28 e 49.
- BATISTA, G. E. A. P. A.; KEOGH, E. J.; TATAW, O. M.; SOUZA, V. M. A. CID: an efficient complexity-invariant distance for time series. **Data Mining and Knowledge Discovery**, v. 28, n. 3, p. 634–669, 2013. ISSN 1573-756X. Disponível em: <<http://dx.doi.org/10.1007/s10618-013-0312-3>>. Citado nas páginas 44, 45, 46 e 197.
- BATISTA, G. E. A. P. A.; MONARD, M. C. An analysis of four missing data treatment methods for supervised learning. **Applied Artificial Intelligence**, v. 17, n. 5, p. 519–533, 2003. [Http://www.icmc.usp.br/gbatista](http://www.icmc.usp.br/gbatista). Citado na página 66.
- BERNARDINI, F. C. **Combinação de Classificadores Utilizando Medidas de Regras de Conhecimento e Algoritmos Genéticos**. Tese (Tese de Doutorado) — ICMC-USP, São Carlos - SP, 2006. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/55/55134/tde-29092006-110806/>>. Citado nas páginas 172 e 173.
- BERNDT, D.; CLIFFORD, J. Using dynamic time warping to find patterns in time series. In: **KDD workshop**. [S.l.: s.n.], 1994. v. 10, n. 16, p. 359–370. Citado na página 118.
- BLOMME, J.; DEBOSSCHER, J.; RIDDER, J. D.; AERTS, C.; GILLILAND, R. L.; CHRISTENSEN-DALSGAARD, J.; KJELDSSEN, H.; BROWN, T. M.; BORUCKI, W. J.; KOCH, D. *et al.* Automated classification of variable stars in the asteroseismology program of the kepler space mission. **The Astrophysical Journal Letters**, IOP Publishing, v. 713, n. 2, p. L204, 2010. Citado na página 48.
- BOLLEN, J.; MAO, H.; ZENG, X. Twitter mood predicts the stock market. **Journal of Computational Science**, v. 2, n. 1, p. 1 – 8, 2011. ISSN 1877-7503. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S187775031100007X>>. Citado na página 51.
- BRESOLIN, A. d. A.; NETO, A.; ALSINA, P. Digit recognition using wavelet and svm in brazilian portuguese. In: IEEE. **Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on**. [S.l.], 2008. p. 1545–1548. Citado nas páginas 28, 47 e 80.
- BROCKWELL, P. J.; DAVIS, R. A. **Introduction to time series and forecasting**. 2. ed. [S.l.]: Springer Science & Business Media, 2002. Citado na página 156.
- CARDONE, A.; GUPTA, S. K.; KARNIK, M. A survey of shape similarity assessment algorithms for product design and manufacturing applications. **Journal of Computing and Information Science in Engineering**, American Society of Mechanical Engineers, v. 3, n. 2, p. 109–118, 2003. Citado na página 42.

- CARRASCAL, U. M. **Contributions to time series data mining departing from the problem of road travel time modeling**. Tese (Doutorado) — University of the Basque Country, 2015. Citado na página 51.
- CAVALCANTE, L. S. **Amostragem e medidas de qualidade de shapelets**. Tese (Doutorado) — Universidade de São Paulo, 2016. Citado na página 81.
- CHA, S. Comprehensive survey on distance/similarity measures between probability density functions. **International Journal of Mathematical Models and Methods in Applied Sciences**, Citeseer, v. 1, n. 4, p. 300–307, 2007. Citado nas páginas 112 e 131.
- CHAN, K.-P.; FU, A.-C. Efficient time series matching by wavelets. In: **Data Engineering, 1999. Proceedings., 15th International Conference on**. [S.l.: s.n.], 1999. p. 126–133. ISSN 1063-6382. Citado nas páginas 28 e 147.
- CHANDOLA, V.; CHEBOLI, D.; KUMAR, V. **Detecting Anomalies in a Time Series Database**. [S.l.], 2009. Citado nas páginas 25 e 53.
- CHANG, C.-C.; LIN, C.-J. Libsvm: a library for support vector machines. **ACM Transactions on Intelligent Systems and Technology (TIST)**, ACM, v. 2, n. 3, p. 27, 2011. Citado nas páginas 166 e 167.
- CHAOVALIT, P.; GANGOPADHYAY, A.; KARABATIS, G.; CHEN, Z. Discrete wavelet transform-based time series analysis and mining. **ACM Computing Surveys (CSUR)**, ACM, v. 43, n. 2, p. 6, 2011. Citado na página 191.
- CHAOVALITWONGSE, W. A.; FAN, Y.-J.; SACHDEO, R. C. On the time series k -nearest neighbor classification of abnormal brain activity. **IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans**, IEEE, v. 37, n. 6, p. 1005–1016, 2007. Citado na página 101.
- CHATFIELD, C. Introduction. In: **The Analysis of Time Series: An Introduction**. 6. ed. New York, USA: Chapman & Hall/CRC, 2004. cap. 1, p. 1. Citado na página 34.
- CHEN, L.; ÖZSU, M. T.; ORIA, V. Robust and fast similarity search for moving object trajectories. In: **ACM. Proceedings of the 2005 ACM SIGMOD international conference on Management of data**. [S.l.], 2005. p. 491–502. Citado na página 128.
- CHEN, Q.; CHEN, L.; LIAN, X.; LIU, Y.; YU, J. X. Indexable pla for efficient similarity search. In: **Proceedings of the 33rd international conference on Very large data bases**. VLDB Endowment, 2007. (VLDB '07), p. 435–446. ISBN 978-1-59593-649-3. Disponível em: <<http://dl.acm.org/citation.cfm?id=1325851.1325903>>. Citado na página 28.
- CHEN, Y.; KEOGH, E.; HU, B.; BEGUM, N.; BAGNALL, A.; MUEEN, A.; BATISTA, G. **The UCR Time Series Classification Archive**. 2015. <www.cs.ucr.edu/~eamonn/time_series_data/>. Citado nas páginas 54 e 56.
- CONDON, J. J.; RANSOM, S. M. Fourier transforms. In: **Essential Radio Astronomy**. [S.l.]: Princenton University Press, 2016. cap. A, p. 281. Citado na página 144.
- COOK, M. J.; O'BRIEN, T. J.; BERKOVIC, S. F.; MURPHY, M.; MOROKOFF, A.; FABINYI, G.; D'SOUZA, W.; YERRA, R.; ARCHER, J.; LITEWKA, L. *et al.* Prediction of seizure likelihood with a long-term, implanted seizure advisory system in patients with drug-resistant

epilepsy: a first-in-man study. **The Lancet Neurology**, Elsevier, v. 12, n. 6, p. 563–571, 2013. Citado na página 48.

COVER, T.; HART, P. Nearest neighbor pattern classification. **Information Theory, IEEE Transactions on**, IEEE, v. 13, n. 1, p. 21–27, 1967. Citado nas páginas 26, 74, 75, 158 e 179.

DAS, G.; GUNOPULOS, D.; MANNILA, H. Finding similar time series. In: SPRINGER. **European Symposium on Principles of Data Mining and Knowledge Discovery**. [S.l.], 1997. p. 88–100. Citado na página 126.

DAVIS, L.; THEOBALD, B.-J.; TOMS, A.; BAGNALL, A. On the extraction and classification of hand outlines. In: _____. **Intelligent Data Engineering and Automated Learning - IDEAL 2011: 12th International Conference, Norwich, UK, September 7-9, 2011. Proceedings**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. p. 92–99. ISBN 978-3-642-23878-9. Disponível em: <http://dx.doi.org/10.1007/978-3-642-23878-9_12>. Citado na página 58.

DEBOSSCHER, J.; SARRO, L.; AERTS, C.; CUYPERS, J.; VANDENBUSSCHE, B.; GARRIDO, R.; SOLANO, E. Automated supervised classification of variable stars-i. methodology. **Astronomy & Astrophysics**, EDP Sciences, v. 475, n. 3, p. 1159–1183, 2007. Citado na página 48.

DEMSAR, J. Statistical comparisons of classifiers over multiple data sets. **The Journal of Machine Learning Research**, JMLR. org, v. 7, p. 1–30, 2006. Citado nas páginas 94, 95, 96, 97 e 98.

DENG, H.; RUNGER, G.; TUV, E.; VLADIMIR, M. A time series forest for classification and feature extraction. **Information Sciences**, v. 239, n. 0, p. 142 – 153, 2013. ISSN 0020-0255. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0020025513001473>>. Citado nas páginas 77 e 175.

DIETTERICH, T. G. Approximate statistical tests for comparing supervised classification learning algorithms. **Neural computation**, MIT Press, v. 10, n. 7, p. 1895–1923, 1998. Citado nas páginas 89 e 94.

DING, H.; TRAJCEVSKI, G.; SCHEUERMANN, P.; WANG, X.; KEOGH, E. Querying and mining of time series data: experimental comparison of representations and distance measures. **Proc. VLDB Endow.**, VLDB Endowment, v. 1, n. 2, p. 1542–1552, ago. 2008. ISSN 2150-8097. Disponível em: <<http://dx.doi.org/10.1145/1454159.1454226>>. Citado nas páginas 26, 76, 158 e 191.

DING, Y.; YANG, X.; LI, J.; KAVS, A. J. A new representation and distance measure for financial time series. In: IEEE. **Information and Financial Engineering (ICIFE), 2010 2nd IEEE International Conference on**. [S.l.], 2010. p. 220–224. Citado na página 189.

ESLING, P.; AGON, C. Time-series data mining. **ACM Computing Surveys (CSUR)**, ACM, v. 45, n. 1, p. 12, 2012. Citado na página 52.

EUACHONGPRASIT, W.; RATANAMAHATANA, C. A. Accurate and efficient retrieval of multimedia time series data under uniform scaling and time warping. In: SPRINGER. **Pacific-Asia Conference on Knowledge Discovery and Data Mining**. [S.l.], 2008. p. 100–111. Citado na página 43.

FAHIMAN, F.; BEZDEK, J. C.; ERFANI, S. M.; LECKIE, C.; PALANISWAMI, M. Fuzzy c-shape: A new algorithm for clustering finite time series waveforms. **arXiv preprint arXiv:1608.01072**, 2016. Citado na página 57.

FALOUTSOS, C.; RANGANATHAN, M.; MANOLOPOULOS, Y. Fast subsequence matching in time-series databases. In: **Proceedings of the 1994 ACM SIGMOD international conference on Management of data**. New York, NY, USA: ACM, 1994. (SIGMOD '94), p. 419–429. ISBN 0-89791-639-5. Disponível em: <<http://doi.acm.org/10.1145/191839.191925>>. Citado na página 141.

FARD, M. J.; PANDYA, A. K.; CHINNAM, R. B.; KLEIN, M. D.; ELLIS, R. D. Distance-based time series classification approach for task recognition with application in surgical robot autonomy. **The International Journal of Medical Robotics and Computer Assisted Surgery**, Wiley Online Library, 2016. Citado na página 80.

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 na página 63.

FERRERO, C. A. **Algoritmo kNN para previsão de dados temporais: funções de previsão e critérios de seleção de vizinhos próximos aplicados a variáveis ambientais em limnologia**. Dissertação (Mestrado) — University of São Paulo, Instituto de Ciências Matemáticas e de Computação, São Carlos, São Paulo, Brazil, 2009. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/55/55134/tde-19052009-135128/>>. Citado na página 51.

FISHER, R. S.; ACEVEDO, C.; ARZIMANOGLU, A.; BOGACZ, A.; CROSS, J. H.; ELGER, C. E.; ENGEL, J.; FORSGREN, L.; FRENCH, J. A.; GLYNN, M. *et al.* Ilae official report: a practical clinical definition of epilepsy. **Epilepsia**, Wiley Online Library, v. 55, n. 4, p. 475–482, 2014. Citado na página 48.

FIX, E.; HODGES, J. **Discriminatory analysis, nonparametric discrimination**. USAF School of Aviation Medicine, Randolph Field, Tex. [S.l.], 1951. Citado na página 74.

FIX, E.; HODGES J. L., J. Discriminatory analysis. nonparametric discrimination: Consistency properties. **International Statistical Review / Revue Internationale de Statistique**, International Statistical Institute (ISI), v. 57, n. 3, p. pp. 238–247, 1989. ISSN 03067734. Originalmente publicado como relatório técnico em 1951, re-editado e republicado em 1989. Disponível em: <<http://www.jstor.org/stable/1403797>>. Citado na página 74.

FLACH, P. **Machine learning: the art and science of algorithms that make sense of data**. [S.l.]: Cambridge University Press, 2012. Citado nas páginas 174 e 196.

FU, T.-c. A review on time series data mining. **Engineering Applications of Artificial Intelligence**, Elsevier, v. 24, n. 1, p. 164–181, 2011. Citado nas páginas 33, 76 e 149.

GANCHEV, T.; FAKOTAKIS, N.; KOKKINAKIS, G. Comparative evaluation of various mfcc implementations on the speaker verification task. In: **Proceedings of the SPECOM**. [S.l.: s.n.], 2005. v. 1, p. 191–194. Citado na página 79.

GENOLINI, C.; ECOCHARD, R.; BENGHEZAL, M.; DRISS, T.; ANDRIEU, S.; SUBTIL, F. kmlshape: An efficient method to cluster longitudinal data (time-series) according to their shapes. **PloS one**, Public Library of Science, v. 11, n. 6, p. e0150738, 2016. Citado nas páginas 52 e 57.

GEURTS, P. Pattern extraction for time series classification. In: RAEDT, L.; SIEBES, A. (Ed.). **Principles of Data Mining and Knowledge Discovery**. Springer Berlin Heidelberg, 2001, (Lecture Notes in Computer Science, v. 2168). p. 115–127. ISBN 978-3-540-42534-2. Disponível em: http://dx.doi.org/10.1007/3-540-44794-6_10. Citado na página 101.

GEURTS, P. **Contributions to decision tree induction: bias/variance tradeoff and time series classification**. Tese (Doutorado) — University of Liège, Belgium, May 2002. Disponível em: <http://www.montefiore.ulg.ac.be/services/stochastic/pubs/2002/Geu02>. Citado na página 76.

GILES, C.; LAWRENCE, S.; TSOI, A. Noisy time series prediction using recurrent neural networks and grammatical inference. **Machine Learning**, Kluwer Academic Publishers, v. 44, n. 1-2, p. 161–183, 2001. ISSN 0885-6125. Disponível em: <http://dx.doi.org/10.1023/A%3A1010884214864>. Citado na página 105.

GIUSTI, R. **Site da Tese de Doutorado**. 2017. Disponível em: <http://sites.labic.icmc.usp.br/rg/tese/>. Citado nas páginas 30, 102, 162, 164, 165 e 195.

GIUSTI, R.; BATISTA, G. An empirical comparison of dissimilarity measures for time series classification. In: **2nd Brazilian International Conference on Intelligent Systems (BRACIS)**. [S.l.: s.n.], 2013. p. 82–88. Citado nas páginas 129 e 130.

GIUSTI, R.; SILVA, D.; BATISTA, G. Time series classification with representation ensembles. In: FROMONT, E.; BIE, T. D.; LEEUWEN, M. van (Ed.). **Advances in Intelligent Data Analysis XIV**. Springer International Publishing, 2015, (Lecture Notes in Computer Science, v. 9385). p. 108–119. ISBN 978-3-319-24464-8. Disponível em: http://dx.doi.org/10.1007/978-3-319-24465-5_10. Citado nas páginas 176 e 184.

GIUSTI, R.; SILVA, D. F.; BATISTA, G. E. Improved time series classification with representation diversity and svm. In: IEEE. **Machine Learning and Applications (ICMLA), 2016 15th IEEE International Conference on**. [S.l.], 2016. p. 1–6. Citado nas páginas 158, 161, 166 e 167.

GORDON, D.; HENDLER, D.; ROKACH, L. Fast randomized model generation for shapelet-based time series classification. **arXiv preprint arXiv:1209.5038**, 2012. Citado nas páginas 81 e 82.

HALL, M.; FRANK, E.; HOLMES, G.; PFAHRINGER, B.; REUTEMANN, P.; WITTEN, I. H. The weka data mining software: An update. **SIGKDD Explor. Newsl.**, ACM, New York, NY, USA, v. 11, n. 1, p. 10–18, nov. 2009. ISSN 1931-0145. Disponível em: <http://doi.acm.org/10.1145/1656274.1656278>. Citado na página 166.

HAMMING, R. W. **Art of Probability**. [S.l.]: Addison Wesley Publishing Company, 1991. Citado na página 114.

HAMOONI, H.; MUEEN, A. Dual-domain hierarchical classification of phonetic time series. In: IEEE. **Data Mining (ICDM), 2014 IEEE International Conference on**. [S.l.], 2014. p. 160–169. Citado nas páginas 47 e 59.

HAN, J.; KAMBER, M. **Data Mining: Concepts and Techniques**. [S.l.]: Morgan kaufmann, 2006. Citado nas páginas 74 e 88.

- HAN, J.; PEI, J.; KAMBER, M. **Data mining: concepts and techniques**. [S.l.]: Elsevier, 2011. Citado na página 52.
- HAO, Y.; CAMPANA, B. J.; KEOGH, E. J. Monitoring and mining insect sounds in visual space. In: **SDM**. [S.l.: s.n.], 2012. p. 792–803. Citado nas páginas 28 e 47.
- HAUSSLER, D. Quantifying inductive bias: AI learning algorithms and valiant's learning framework. **Artificial intelligence**, Elsevier, v. 36, n. 2, p. 177–221, 1988. Citado na página 66.
- HEARN, D. J. Shape analysis for the automated identification of plants from images of leaves. **Taxon**, International Association for Plant Taxonomy, v. 58, n. 3, p. 934–954, 2009. Citado na página 50.
- HILLS, J.; LINES, J.; BARANAUSKAS, E.; MAPP, J.; BAGNALL, A. Classification of time series by shapelet transformation. **Data Mining and Knowledge Discovery**, Springer US, v. 28, n. 4, p. 851–881, 2014. ISSN 1384-5810. Disponível em: <<http://dx.doi.org/10.1007/s10618-013-0322-1>>. Citado na página 50.
- HOULE, M. E.; KRIEGLER, H.-P.; KRÖGER, P.; SCHUBERT, E.; ZIMEK, A. Can shared-neighbor distances defeat the curse of dimensionality? In: **Proceedings of the 21st International Conference on Scientific and Statistical Database Management (SSDBM)**. Heidelberg, Germany: Springer, 2010. Citado na página 26.
- HU, B.; CHEN, Y.; KEOGH, E. Time series classification under more realistic assumptions. In: _____. **Proceedings of the 2013 SIAM International Conference on Data Mining**. [s.n.], 2013. cap. 63, p. 578–586. Disponível em: <<http://epubs.siam.org/doi/abs/10.1137/1.9781611972832.64>>. Citado nas páginas 37 e 43.
- ITAKURA, F. Minimum prediction residual principle applied to speech recognition. **IEEE Transactions on Acoustics, Speech, and Signal Processing**, IEEE, v. 23, n. 1, p. 67–72, 1975. Citado na página 121.
- 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 52.
- JANYALIKIT, T.; SATHIANWIRIYAKHUN, P.; SIVARAKS, H.; RATANAMAHATANA, C. A. An enhanced support vector machine for faster time series classification. In: **Proceedings of the 8th Asian Conference on Intelligent Information and Database Systems**. Da Nang, Vietnam: Springer Berlin Heidelberg, 2016. p. 616–625. ISBN 978-3-662-49381-6. Disponível em: <http://dx.doi.org/10.1007/978-3-662-49381-6_59>. Citado na página 100.
- JEONG, Y.-S.; JEONG, M. K.; OMITAOMU, O. A. Weighted dynamic time warping for time series classification. **Pattern Recognition**, Elsevier, v. 44, n. 9, p. 2231–2240, 2011. Citado nas páginas 122, 123 e 124.
- JOLLIFFE, I. Principal component analysis for time series and other non-independent data. In: **Principal Component Analysis**. 2. ed. Springer New York, 2002, (Springer Series in Statistics). p. 299–337. ISBN 978-0-387-95442-4. Disponível em: <http://dx.doi.org/10.1007/0-387-22440-8_12>. Citado na página 155.
- JURADO, F.; SAENZ, J. R. Comparison between discrete {STFT} and wavelets for the analysis of power quality events. **Electric Power Systems Research**, v. 62, n. 3, p. 183 – 190, 2002. ISSN 0378-7796. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0378779602000354>>. Citado na página 145.

KAM, H.; SUNG, J.; PARK, R. Prediction of daily patient numbers for a regional emergency medical center using time series analysis. **Healthcare informatics research**, v. 16, n. 3, p. 158–165, 2010. Citado na página 51.

KATE, R. Using dynamic time warping distances as features for improved time series classification. **Data Mining and Knowledge Discovery**, Springer US, v. 30, n. 2, p. 283–312, 2016. ISSN 1384-5810. Disponível em: <<http://dx.doi.org/10.1007/s10618-015-0418-x>>. Citado nas páginas 57, 90, 100, 105, 158, 159 e 167.

KEOGH, E.; CELLY, B.; RATANAMAHATANA, C. A.; ZORDAN, V. B. A novel technique for indexing video surveillance data. In: **First ACM SIGMM International Workshop on Video Surveillance**. New York, NY, USA: ACM, 2003. (IWVS '03), p. 98–106. ISBN 1-58113-780-X. Disponível em: <<http://doi.acm.org/10.1145/982452.982465>>. Citado na página 44.

KEOGH, E.; CHAKRABARTI, K.; PAZZANI, M.; MEHROTRA, S. Dimensionality reduction for fast similarity search in large time series databases. **Knowledge and Information Systems**, Springer-Verlag London Limited, v. 3, n. 3, p. 263–286, 2001. ISSN 0219-1377. Disponível em: <<http://dx.doi.org/10.1007/PL00011669>>. Citado nas páginas 28, 149 e 152.

KEOGH, E.; KASETTY, S. On the need for time series data mining benchmarks: a survey and empirical demonstration. **Data Mining and Knowledge Discovery**, Springer, v. 7, n. 4, p. 349–371, 2003. Citado nas páginas 37 e 101.

KEOGH, E.; LIN, J.; FU, A. Hot sax: Efficiently finding the most unusual time series subsequence. In: IEEE. **Fifth IEEE International Conference on Data Mining (ICDM'05)**. [S.l.], 2005. p. 8–pp. Citado nas páginas 53 e 154.

KEOGH, E.; LONARDI, S.; RATANAMAHATANA, C. Towards parameter-free data mining. In: **Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. Seattle, WA: [s.n.], 2004. Citado na página 122.

KEOGH, E.; RATANAMAHATANA, C. A. Exact indexing of dynamic time warping. **Knowledge and Information Systems**, Springer London, v. 7, p. 358–386, 2005. ISSN 0219-1377. 10.1007/s10115-004-0154-9. Disponível em: <<http://dx.doi.org/10.1007/s10115-004-0154-9>>. Citado na página 122.

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: **Proceedings of the 32Nd International Conference on Very Large Data Bases**. VLDB Endowment, 2006. (VLDB '06), p. 882–893. Disponível em: <<http://dl.acm.org/citation.cfm?id=1182635.1164203>>. Citado nas páginas 42 e 50.

KEOGH, E.; WEI, L.; XI, X.; VLACHOS, M.; LEE, S.-H.; PROTOPAPAS, P. Supporting exact indexing of arbitrarily rotated shapes and periodic time series under euclidean and warping distance measures. **The VLDB Journal**, Springer-Verlag New York, Inc., Secaucus, NJ, USA, v. 18, p. 611–630, June 2009. ISSN 1066-8888. Disponível em: <<http://dx.doi.org/10.1007/s00778-008-0111-4>>. Citado nas páginas 35 e 41.

KHANDANI, A. E.; KIM, A. J.; LO, A. W. Consumer credit-risk models via machine-learning algorithms. **Journal of Banking & Finance**, v. 34, n. 11, p. 2767 – 2787, 2010. ISSN 0378-4266. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0378426610002372>>. Citado na página 63.

KIM, K.; LEE, C.; O'LEARY, K.; ROSENAUER, S.; MEHROTRA, S. **Predicting patient volumes in hospital medicine: A comparative study of different time series forecasting methods**. [S.l.], 2014. Citado na página 51.

KIRCHGÄSSNER, G.; WOLTERS, J. Introduction and basics. In: **Introduction to Modern Time Series Analysis**. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg, 2007. cap. 1, p. 9. ISBN 978-3-540-73291-4. Disponível em: <http://aleph.bib.uni-mannheim.de/F/?func=find-b&request=276336909&find_code=020&adjacent=N&local_base=MAN01PUBLIC&x=0&y=0>. Citado na página 34.

_____. **Introduction to Modern Time Series Analysis**. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg, 2007. Online-Ressource p. ISBN 978-3-540-73291-4. Disponível em: <http://aleph.bib.uni-mannheim.de/F/?func=find-b&request=276336909&find_code=020&adjacent=N&local_base=MAN01PUBLIC&x=0&y=0>. Citado na página 78.

KOHAVI, R. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: STANFORD, CA. **Ijcai**. [S.l.], 1995. v. 14, n. 2, p. 1137–1145. Citado na página 88.

KUMAR, N.; LOLLA, N.; KEOGH, E.; LONARDI, S.; RATANAMAHATANA, C. A. Time-series bitmaps: a practical visualization tool for working with large time series databases. In: CITESEER. **SIAM 2005 Data Mining Conference**. [S.l.], 2005. Citado nas páginas 52, 53 e 196.

LAKSHMINARAYAN, K.; HARP, S. A.; SAMAD, T. Imputation of missing data in industrial databases. **Applied Intelligence**, v. 11, p. 259–275, 1999. Citado na página 66.

LAWLER, G. F. Introduction to stochastic processes. In: _____. [S.l.]: Preliminaries, 2006. cap. 0, p. 1–8. Citado na página 34.

LEIS, J. W. **Digital signal processing using MATLAB for students and researchers**. [S.l.]: John Wiley & Sons, 2011. Citado na página 139.

LEMES, C. I. **Algoritmos anytime baseados em instâncias para classificação em fluxo de dados**. Tese (Doutorado) — Universidade de São Paulo, 2016. Citado na página 49.

LEVENSHTEIN, V. I. Binary codes capable of correcting deletions, insertions, and reversals. In: **Soviet physics doklady**. [S.l.: s.n.], 1966. v. 10, n. 8, p. 707–710. Citado na página 127.

LIAO, T. W. Clustering of time series data—a survey. **Pattern Recognition**, v. 38, n. 11, p. 1857–1874, 2005. ISSN 0031-3203. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0031320305001305>>. Citado na página 52.

LICHMAN, M. **UCI Machine Learning Repository**. 2013. Disponível em: <<http://archive.ics.uci.edu/ml>>. Citado na página 60.

LIN, J.; KEOGH, E.; WEI, L.; LONARDI, S. Experiencing sax: a novel symbolic representation of time series. **Data Mining and Knowledge Discovery**, Springer US, v. 15, n. 2, p. 107–144, 2007. ISSN 1384-5810. Disponível em: <<http://dx.doi.org/10.1007/s10618-007-0064-z>>. Citado nas páginas 28, 39, 41, 150, 151, 152 e 154.

- LIN, J.; KHADE, R.; LI, Y. Rotation-invariant similarity in time series using bag-of-patterns representation. **Journal of Intelligent Information Systems**, Springer US, v. 39, n. 2, p. 287–315, 2012. ISSN 0925-9902. Disponível em: <<http://dx.doi.org/10.1007/s10844-012-0196-5>>. Citado nas páginas 52, 174 e 196.
- LINES, J.; BAGNALL, A. Time series classification with ensembles of elastic distance measures. **Data Mining and Knowledge Discovery**, Springer, v. 29, n. 3, p. 565–592, 2015. Citado nas páginas 80 e 175.
- LINES, J.; HILLS, J.; BAGNALL, A. The collective of transformation-based ensembles for time-series classification. 2014. Citado na página 192.
- MALETZKE, A. G. **Uma metodologia para extração de conhecimento em séries temporais por meio da identificação de motivos e da extração de características**. Dissertação (Mestrado) — Instituto de Ciências Matemáticas e de Computação, University of São Paulo., 2009. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/55/55134/tde-04062009-201445/>>. Citado na página 53.
- MALLAT, S. G. A theory for multiresolution signal decomposition: the wavelet representation. **IEEE transactions on pattern analysis and machine intelligence**, Ieee, v. 11, n. 7, p. 674–693, 1989. Citado na página 147.
- MELGANI, F.; BAZI, Y. Classification of electrocardiogram signals with support vector machines and particle swarm optimization. **Information Technology in Biomedicine, IEEE Transactions on**, v. 12, n. 5, p. 667–677, 2008. ISSN 1089-7771. Citado na página 26.
- MITCHELL, T. M. **Machine Learning**. 1. ed. [S.l.]: WCB/McGraw-Hill, 1997. Citado nas páginas 65, 66, 69, 84, 87, 171 e 174.
- MONARD, M. C.; BARANAUSKAS, J. A. Conceitos sobre aprendizado de máquina. In: REZENDE, S. (Ed.). **Sistemas Inteligentes: Fundamentos e Aplicações**. [S.l.]: Editora Manole Ltda, 2003. cap. 4, p. 90–114. Citado na página 84.
- MUEEN, A.; KEOGH, E.; YOUNG, N. Logical-shapelets: an expressive primitive for time series classification. In: ACM. **Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining**. [S.l.], 2011. p. 1154–1162. Citado na página 83.
- MUEEN, A.; KEOGH, E.; ZHU, Q.; CASH, S.; WESTOVER, B. Exact discovery of time series motifs. In: **Proc. of 2009 SIAM International Conference on Data Mining: SDM**. [S.l.: s.n.], 2009. p. 1–12. Citado na página 25.
- MUEEN, A.; NATH, S.; LIU, J. Fast approximate correlation for massive time-series data. In: ACM. **Proceedings of the 2010 ACM SIGMOD International Conference on Management of data**. [S.l.], 2010. p. 171–182. Citado na página 83.
- NANOPOULOS, A.; ALCOCK, R.; MANOLOPOULOS, Y. Feature-based classification of time-series data. **Information processing and technology**, Citeseer, p. 49–61, 2001. Citado nas páginas 78 e 79.
- NAWROCKA, A.; LAMORSKA, J. Determination of food quality by using spectroscopic methods. **Advances in Agrophysical Research**, p. 347–367, 2013. Citado nas páginas 50, 57 e 58.

- NOLL, A. M. Cepstrum pitch determination. **The Journal of the Acoustical Society of America**, ASA, v. 41, n. 2, p. 293–309, 1967. Disponível em: <<http://link.aip.org/link/?JAS/41/293/1>>. Citado na página 47.
- OATES, T.; MACKENZIE, C.; STEIN, D.; STANSBURY, L.; DUBOSE, J.; AARABI, B.; HU, P. Exploiting representational diversity for time series classification. In: IEEE. **Machine Learning and Applications (ICMLA), 2012 11th International Conference on**. [S.l.], 2012. v. 2, p. 538–544. Citado nas páginas 76, 174 e 191.
- ODAN, F. K.; FERRERO, C. A.; REIS, L. F. R.; MONARD, M. C. Análise comparativa dos modelos knn-tsp e série de fourier para previsão de demanda horária para abastecimento de água. **XVIII Simpósio Brasileiro de Recursos Hídricos—ABRH**, 2009. Citado na página 25.
- OLSZEWSKI, R. T. **Generalized feature extraction for structural pattern recognition in time-series data**. [S.l.], 2001. Citado nas páginas 56 e 57.
- PARMEZAN, A. R. S. **Predição de séries temporais por similaridade**. Tese (Doutorado) — Universidade de São Paulo, 2016. Citado nas páginas 46, 50, 156 e 160.
- PARMEZAN, A. R. S.; BATISTA, G. E. A. P. A. **ICMC-USP Time Series Prediction Repository**. 2014. Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo. Disponível em: <http://sites.labicc.icmc.usp.br/icmc_tspr>. Citado na página 143.
- PETROSIAN, A. Kolmogorov complexity of finite sequences and recognition of different preictal eeg patterns. In: IEEE. **Computer-Based Medical Systems, 1995., Proceedings of the Eighth IEEE Symposium on**. [S.l.], 1995. p. 212–217. Citado na página 48.
- POPIVANOV, I.; MILLER, R. Similarity search over time-series data using wavelets. In: **Data Engineering, 2002. Proceedings. 18th International Conference on**. [S.l.: s.n.], 2002. p. 212–221. ISSN 1063-6382. Citado na página 147.
- PRATI, R. C.; BATISTA, G. E. Distância invariante à complexidade baseada em dimensão fractal para classificação de séries temporais. In: **IX Encontro Nacional de Inteligência Artificial (ENIA-2012)**. [S.l.: s.n.], 2012. p. 1–12. Citado nas páginas 46, 76 e 130.
- PRATI, R. C.; BATISTA, G. E. A. P. A. A complexity invariant measure based on fractal dimension for time series classification. **International Journal of Natural Computing Research**, 2013. In press. Citado na página 44.
- PROTOPAPAS, P.; GIAMMARCO, J.; FACCIOLI, L.; STRUBLE, M.; DAVE, R.; ALCOCK, C. Finding outlier light curves in catalogues of periodic variable stars. **Monthly Notices of the Royal Astronomical Society**, Oxford University Press, v. 369, n. 2, p. 677–696, 2006. Citado nas páginas 41 e 53.
- QUINLAN, J. R. **C4.5: programs for machine learning**. [S.l.]: Morgan Kaufmann, 2014. (Morgan Kaufmann series in machine learning). Citado nas páginas 67 e 72.
- RATANAMAHATANA, C.; KEOGH, E. Everything you know about dynamic time warping is wrong. In: **Third Workshop on Mining Temporal and Sequential Data**. [S.l.: s.n.], 2004. p. 22–25. Citado na página 120.
- RATANAMAHATANA, C. A.; KEOGH, E. J. Making time-series classification more accurate using learned constraints. In: BERRY, M. W.; DAYAL, U.; KAMATH, C.; SKILLICORN, D. B. (Ed.). **SDM**. [S.l.]: SIAM, 2004. ISBN 0-89871-568-7. Citado na página 121.

- RATH, T. M.; MANMATHA, R. Word image matching using dynamic time warping. In: **Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on**. [S.l.: s.n.], 2003. v. 2, p. II-521-II-527 vol.2. ISSN 1063-6919. Citado nas páginas 43 e 50.
- REYNOLDS, D. Gaussian mixture models. **Encyclopedia of biometrics**, Springer, p. 827-832, 2015. Citado na página 48.
- REZENDE, S. (Ed.). **Sistemas inteligentes: fundamentos e aplicações**. [S.l.]: Editora Manole Ltda, 2003. Citado na página 52.
- RICH, E.; KNIGHT, K. Problems, problem spaces, and search. In: _____. 2nd. ed. [S.l.]: McGraw-Hill, 1991. cap. 2, p. 54. Citado na página 65.
- RICHARDS, J. W.; STARR, D. L.; BUTLER, N. R.; BLOOM, J. S.; BREWER, J. M.; CRELLIN-QUICK, A.; HIGGINS, J.; KENNEDY, R.; RISCHARD, M. On machine-learned classification of variable stars with sparse and noisy time-series data. **The Astrophysical Journal**, v. 733, n. 1, p. 10, 2011. Disponível em: <<http://stacks.iop.org/0004-637X/733/i=1/a=10>>. Citado nas páginas 26 e 48.
- ROKACH, L. A survey of clustering algorithms. In: **The Data Mining and Knowledge Discovery Handbook**. 2. ed. [S.l.]: Springer, 2005. cap. 14. Citado na página 25.
- _____. Ensemble-based classifiers. **Artificial Intelligence Review**, Springer, v. 33, n. 1, p. 1-39, 2010. Citado na página 173.
- RUSSELL, S.; NORVIG, P. **Artificial Intelligence**. 4. ed. [S.l.]: Pearson Education, 2010. Citado nas páginas 65, 70, 159 e 196.
- SAINI, I.; SINGH, D.; KHOSLA, A. {QRS} detection using k-nearest neighbor algorithm (knn) and evaluation on standard {ECG} databases. **Journal of Advanced Research**, v. 4, n. 4, p. 331 - 344, 2013. ISSN 2090-1232. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S209012321200046X>>. Citado na página 49.
- SAKOE, H.; CHIBA, S. Dynamic programming algorithm optimization for spoken word recognition. **IEEE Transactions on Acoustics, Speech, and Signal Processing**, v. 26, p. 43-49, 1978. Citado nas páginas 117 e 120.
- SCHÄFER, P. The boss is concerned with time series classification in the presence of noise. **Data Mining and Knowledge Discovery**, v. 29, n. 6, p. 1505-1530, 2015. ISSN 1573-756X. Disponível em: <<http://dx.doi.org/10.1007/s10618-014-0377-7>>. Citado na página 191.
- SCHUSTER, A. The periodicity of sun-spots. **The Astrophysical Journal**, v. 23, p. 101, 1906. Citado na página 34.
- SHOKOOHI-YEKTA, M.; CHEN, Y.; CAMPANA, B.; HU, B.; ZAKARIA, J.; KEOGH, E. Discovery of meaningful rules in time series. In: **Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. New York, NY, USA: ACM, 2015. (KDD '15), p. 1085-1094. ISBN 978-1-4503-3664-2. Disponível em: <<http://doi.acm.org/10.1145/2783258.2783306>>. Citado na página 100.
- SILVA, D. F. **Classificação de séries temporais por similaridade e extração de atributos com aplicação na identificação automática de insetos**. Dissertação (Mestrado) — Universidade de São Paulo, 2014. Citado nas páginas 26, 119 e 121.

SILVA, D. F.; SOUZA, V. M. de; BATISTA, G. E.; GIUSTI, R. Spoken digit recognition in portuguese using line spectral frequencies. In: **Advances in Artificial Intelligence–IBERAMIA 2012**. [S.l.]: Springer, 2012. p. 241–250. Citado na página [26](#).

SILVERMAN, H.; DIXON, N. A comparison of several speech-spectra classification methods. **Acoustics, Speech and Signal Processing, IEEE Transactions on**, v. 24, n. 4, p. 289–295, 1976. ISSN 0096-3518. Citado na página [25](#).

SMITH, S. Eeg in the diagnosis, classification, and management of patients with epilepsy. **Journal of Neurology, Neurosurgery & Psychiatry**, BMJ Publishing Group Ltd, v. 76, n. suppl 2, p. ii2–ii7, 2005. Citado na página [48](#).

SOKOLOVA, M.; LAPALME, G. A systematic analysis of performance measures for classification tasks. **Information Processing & Management**, Elsevier, v. 45, n. 4, p. 427–437, 2009. Citado na página [86](#).

SOUZA, V. M. A. d. **Classificação de fluxo de dados não estacionários com aplicação em sensores identificadores de insetos**. Tese (Doutorado) — Universidade de São Paulo, 2016. Citado nas páginas [49](#) e [53](#).

STARNER, T.; MAKHOUL, J.; SCHWARTZ, R.; CHOU, G. On-line cursive handwriting recognition using speech recognition methods. In: IEEE. **Acoustics, Speech, and Signal Processing, 1994. ICASSP-94., 1994 IEEE International Conference on**. [S.l.], 1994. p. V–125. Citado na página [26](#).

TAN, P.-N.; STEINBACH, M.; KUMAR, V. **Introduction to data mining**. [S.l.]: Addison-Wesley, 2006. Citado na página [155](#).

THEODORIDIS, S.; KOUTROUMBAS, K. **Pattern Recognition**. 4. ed. [S.l.]: Else, 2009. Citado na página [159](#).

THOMPSON, W. C. Painting the target around the matching profile: the texas sharpshooter fallacy in forensic dna interpretation. **Law, Prob. & Risk**, HeinOnline, v. 8, p. 257, 2009. Citado na página [92](#).

TZALLAS, A.; TSIPOURAS, M.; FOTIADIS, D. Automatic seizure detection based on time-frequency analysis and artificial neural networks. **Computational Intelligence and Neuroscience**, Hindawi Publishing Corporation, v. 2007, 2007. Citado na página [48](#).

UENO, K.; XI, X.; KEOGH, E.; LEE, D.-J. Anytime classification using the nearest neighbor algorithm with applications to stream mining. In: IEEE. **Data Mining, 2006. ICDM'06. Sixth International Conference on**. [S.l.], 2006. p. 623–632. Citado na página [183](#).

UZINSKI, J. C. Momentos nulos e regularidade wavelet na detecção de falhas em sinais. Universidade Estadual Paulista (UNESP), 2013. Citado na página [148](#).

VIJAYARANI, S.; ILAMATHI, M. J.; NITHYA, M. Preprocessing techniques for text mining-an overview. **International Journal of Computer Science & Communication Networks**, v. 5, n. 1, p. 7–16, 2015. Citado na página [64](#).

VLACHOS, M.; KOLLIOS, G.; GUNOPULOS, D. Discovering similar multidimensional trajectories. In: **Data Engineering, 2002. Proceedings. 18th International Conference on**. [S.l.: s.n.], 2002. p. 673–684. ISSN 1063-6382. Citado na página [126](#).

WANG, J.; LIU, P.; SHE, M. F.; NAHAVANDI, S.; KOUZANI, A. Bag-of-words representation for biomedical time series classification. **Biomedical Signal Processing and Control**, v. 8, n. 6, p. 634 – 644, 2013. ISSN 1746-8094. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S174680941300089X>>. Citado na página 48.

WANG, L.; WANG, Z.; LIU, S. An effective multivariate time series classification approach using echo state network and adaptive differential evolution algorithm. **Expert Systems with Applications**, Elsevier, v. 43, p. 237–249, 2016. Citado na página 57.

WANG, X.; MUEEN, A.; DING, H.; TRAJCEVSKI, G.; SCHEUERMANN, P.; KEOGH, E. Experimental comparison of representation methods and distance measures for time series data. **Data Mining and Knowledge Discovery**, Springer US, v. 26, p. 275–309, 2013. ISSN 1384-5810. Disponível em: <<http://dx.doi.org/10.1007/s10618-012-0250-5>>. Citado nas páginas 26 e 152.

WEI, L.; KEOGH, E.; XI, X. SAXually explicit images: finding unusual shapes. In: IEEE. **Sixth International Conference on Data Mining (ICDM'06)**. [S.l.], 2006. p. 711–720. Citado na página 50.

WEISS, S. M.; KULIKOWSKI, C. A. **Computer Systems That Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1991. ISBN 1-55860-065-5. Citado nas páginas 85 e 88.

WIJK, J. J. v.; SELOW, E. R. V. Cluster and calendar based visualization of time series data. In: IEEE. **Information Visualization, 1999.(Info Vis'99) Proceedings. 1999 IEEE Symposium on**. [S.l.], 1999. p. 4–9. Citado na página 52.

WILLEMS, J. L.; ARNAUD, P.; BEMMEL, J. H. V.; BOURDILLON, P. J.; DEGANI, R.; DENIS, B.; GRAHAM, I.; HARMS, F. M.; MACFARLANE, P. W.; MAZZOCCA, G. *et al.* A reference data base for multilead electrocardiographic computer measurement programs. **Journal of the American College of Cardiology**, Elsevier, v. 10, n. 6, p. 1313–1321, 1987. Citado na página 49.

WU, X.; KUMAR, V.; QUINLAN, J. R.; GHOSH, J.; YANG, Q.; MOTODA, H.; MCLACHLAN, G. J.; NG, A.; LIU, B.; PHILIP, S. Y. *et al.* Top 10 algorithms in data mining. **Knowledge and Information Systems**, Springer, v. 14, n. 1, p. 1–37, 2008. Citado na página 76.

XI, X.; KEOGH, E.; SHELTON, C.; WEI, L.; RATANAMAHATANA, C. A. Fast time series classification using numerosity reduction. In: **In ICML'06**. [S.l.: s.n.], 2006. p. 1033–1040. Citado na página 122.

XI, X.; KEOGH, E.; WEI, L.; MAFRA-NETO, A. Finding motifs in a database of shapes. In: _____. **Proceedings of the 2007 SIAM International Conference on Data Mining**. [s.n.], 2007. cap. 23, p. 249–260. Disponível em: <<http://epubs.siam.org/doi/abs/10.1137/1.9781611972771.23>>. Citado na página 50.

XIAO, Q.; LUO, Y.; GAO, S. Human motion retrieval with symbolic aggregate approximation. In: IEEE. **Control and Decision Conference (CCDC), 2012 24th Chinese**. [S.l.], 2012. p. 3632–3636. Citado na página 52.

YANG, Y.; DENG, Q.; SHEN, F.; ZHAO, J.; LUO, C. A shapelet learning method for time series classification. In: IEEE. **Tools with Artificial Intelligence (ICTAI), 2016 IEEE 28th International Conference on**. [S.l.], 2016. p. 423–430. Citado na página 81.

YANKOV, D.; KEOGH, E.; MEDINA, J.; CHIU, B.; ZORDAN, V. Detecting time series motifs under uniform scaling. In: ACM. **Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining**. [S.l.], 2007. p. 844–853. Citado na página 53.

YANKOV, D.; KEOGH, E.; REBBAPRAGADA, U. Disk aware discord discovery: Finding unusual time series in terabyte sized datasets. In: **Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on**. [S.l.: s.n.], 2007. p. 381–390. ISSN 1550-4786. Citado na página 25.

_____. Disk aware discord discovery: finding unusual time series in terabyte sized datasets. **Knowledge and Information Systems**, Springer, v. 17, n. 2, p. 241–262, 2008. Citado na página 54.

YE, L.; KEOGH, E. Time series shapelets: a new primitive for data mining. In: ACM. **Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining**. [S.l.], 2009. p. 947–956. Citado nas páginas 81 e 83.

_____. Time series shapelets: a novel technique that allows accurate, interpretable and fast classification. **Data Mining and Knowledge Discovery**, Springer US, v. 22, n. 1-2, p. 149–182, 2011. ISSN 1384-5810. Disponível em: <<http://dx.doi.org/10.1007/s10618-010-0179-5>>. Citado nas páginas 37 e 196.

YE, L.; WANG, X.; KEOGH, E. J.; MAFRA-NETO, A. Autocannibalistic and anyspace indexing algorithms with application to sensor data mining. In: SIAM. **SDM**. [S.l.], 2009. p. 85–96. Citado na página 108.

YI, B.-K.; FALOUTSOS, C. Fast time sequence indexing for arbitrary lp norms. In: VLDB. [S.l.], 2000. Citado nas páginas 149 e 152.

ZHENG, Y.; LIU, Q.; CHEN, E.; GE, Y.; ZHAO, J. L. Time series classification using multi-channels deep convolutional neural networks. In: SPRINGER. **International Conference on Web-Age Information Management**. [S.l.], 2014. p. 298–310. Citado na página 105.

ZHENG, Z.; LOW, B. T. Classifying unseen cases with many missing values. In: **Pacific-Asia Conference on Knowledge Discovery and Data Mining**. [S.l.: s.n.], 1999. p. 370–374. Citado na página 66.

ZHOU, Y.; LIU, Y.; YANG, J.; HE, X.; LIU, L. A taxonomy of label ranking algorithms. **Journal of Computers**, v. 9, n. 3, p. 557–565, 2014. Citado na página 180.

APÊNDICE: RESULTADOS ESTENDIDOS

A.1 Viés e variância dos métodos de reamostragem

Os resultados seguintes referem-se ao experimento discutido na Seção 3.9.1. Cada *boxplot* refere-se a um conjunto de dados do repositório UCR. Os valores representados nos gráficos são o escore de enviesamento do método de reamostragem. As caixas indicam os quartis e os bigodes indicam valores máximos e mínimos, exceto *outliers*, representados pelo símbolo “+”.

A.2 Resultados completos do teste de normalização e *ranks* de Friedman

A tabela a seguir refere-se à totalidade dos resultados experimentais do teste discutido na Seção 3.9.3. Os valores de *ranks* apresentados a seguir dizem respeito aos *ranks* atribuídos aos diferentes métodos pelo teste de hipótese de Friedman. Os resultados não permitem rejeitar a hipótese nula de que os algoritmos são idênticos com $\alpha = 0,05$ ou mesmo com $\alpha = 0,10$.

Conjunto	Acurácia (<i>rank</i>)					
	Z-norm.		Interv. [0, 1]		Histograma	
50Words	0,631	(3,0)	0,653	(1,0)	0,638	(2,0)
Adiac	0,611	(2,0)	0,618	(1,0)	0,592	(3,0)

ArrowHead	0,800	(3,0)	0,833	(2,0)	0,861	(1,0)
Beef	0,667	(1,0)	0,400	(3,0)	0,500	(2,0)
BeetleFly	0,750	(1,0)	0,650	(2,5)	0,650	(2,5)
BirdChicken	0,550	(3,0)	0,800	(1,5)	0,800	(1,5)
Car	0,733	(1,0)	0,700	(3,0)	0,717	(2,0)
CBF	0,852	(2,0)	0,933	(1,0)	0,833	(3,0)
ChlorineConcentration	0,650	(1,0)	0,606	(3,0)	0,625	(2,0)
CinCECGtorso	0,897	(1,0)	0,750	(3,0)	0,825	(2,0)
Coffee	1,000	(2,0)	1,000	(2,0)	1,000	(2,0)
Computers	0,576	(3,0)	0,604	(2,0)	0,632	(1,0)
CricketX	0,577	(3,0)	0,636	(1,0)	0,579	(2,0)
CricketY	0,567	(1,0)	0,536	(2,0)	0,510	(3,0)
CricketZ	0,587	(2,0)	0,592	(1,0)	0,541	(3,0)
DiatomSizeReduction	0,935	(3,0)	0,938	(1,5)	0,938	(1,5)
DistalPhalanxOutlineAgeGroup	0,782	(1,0)	0,676	(2,0)	0,662	(3,0)
DistalPhalanxOutlineCorrect	0,752	(1,0)	0,674	(3,0)	0,681	(2,0)
DistalPhalanxTW	0,728	(1,0)	0,612	(3,0)	0,619	(2,0)
Earthquakes	0,674	(2,0)	0,734	(1,0)	0,669	(3,0)
ECG200	0,880	(1,0)	0,820	(3,0)	0,830	(2,0)
ECG5000	0,925	(3,0)	0,932	(1,5)	0,932	(1,5)
ECGFiveDays	0,797	(1,0)	0,652	(3,0)	0,783	(2,0)
ElectricDevices	0,551	(3,0)	0,703	(1,0)	0,683	(2,0)
FaceAll	0,714	(3,0)	0,871	(1,0)	0,870	(2,0)
FaceFour	0,784	(1,0)	0,625	(2,5)	0,625	(2,5)
FacesUCR	0,769	(1,0)	0,730	(2,0)	0,710	(3,0)
Fish	0,783	(1,0)	0,760	(2,0)	0,754	(3,0)
FordA	0,659	(2,0)	0,659	(1,0)	0,657	(3,0)
FordB	0,558	(2,0)	0,558	(1,0)	0,552	(3,0)
GunPoint	0,913	(2,0)	0,920	(1,0)	0,900	(3,0)
Ham	0,600	(3,0)	0,780	(2,0)	0,807	(1,0)
HandOutlines	0,801	(3,0)	0,862	(1,0)	0,859	(2,0)
Haptics	0,370	(3,0)	0,477	(2,0)	0,503	(1,0)
Herring	0,516	(1,0)	0,375	(3,0)	0,422	(2,0)
InlineSkate	0,342	(1,0)	0,290	(3,0)	0,330	(2,0)
InsectWingbeatSound	0,562	(1,0)	0,536	(2,0)	0,505	(3,0)
ItalyPowerDemand	0,955	(2,0)	0,955	(3,0)	0,970	(1,0)
LargeKitchenAppliances	0,493	(3,0)	0,592	(1,0)	0,555	(2,0)
Lightning2	0,754	(1,0)	0,733	(2,0)	0,650	(3,0)
Lightning7	0,575	(2,0)	0,671	(1,0)	0,543	(3,0)
MALLAT	0,914	(3,0)	0,964	(1,0)	0,927	(2,0)
Meat	0,933	(3,0)	1,000	(1,5)	1,000	(1,5)
MedicalImages	0,684	(3,0)	0,703	(2,0)	0,706	(1,0)
MiddlePhalanxOutlineAgeGroup	0,740	(1,0)	0,545	(2,0)	0,519	(3,0)
MiddlePhalanxOutlineCorrect	0,753	(2,0)	0,773	(1,0)	0,732	(3,0)
MiddlePhalanxTW	0,561	(1,0)	0,506	(2,5)	0,506	(2,5)
MoteStrain	0,879	(1,0)	0,650	(2,0)	0,600	(3,0)
NonInvasiveFetalECG1	0,829	(1,0)	0,790	(3,0)	0,819	(2,0)
NonInvasiveFetalECG2	0,880	(1,0)	0,877	(3,0)	0,878	(2,0)
OliveOil	0,867	(1,5)	0,800	(3,0)	0,867	(1,5)
OSULeaf	0,521	(3,0)	0,650	(1,0)	0,625	(2,0)
PhalangesOutlinesCorrect	0,761	(3,0)	0,792	(2,0)	0,795	(1,0)
Phoneme	0,109	(1,0)	0,065	(3,0)	0,084	(2,0)
Plane	0,962	(3,0)	0,981	(1,5)	0,981	(1,5)
ProximalPhalanxOutlineAgeGroup	0,785	(3,0)	0,787	(2,0)	0,797	(1,0)
ProximalPhalanxOutlineCorrect	0,808	(3,0)	0,817	(1,0)	0,813	(2,0)
ProximalPhalanxTW	0,708	(2,0)	0,712	(1,0)	0,702	(3,0)
RefrigerationDevices	0,395	(2,0)	0,493	(1,0)	0,384	(3,0)

ScreenType	0,360 (3,0)	0,453 (1,0)	0,435 (2,0)
ShapeletSim	0,539 (1,0)	0,450 (2,0)	0,300 (3,0)
ShapesAll	0,752 (2,0)	0,748 (3,0)	0,757 (1,0)
SmallKitchenAppliances	0,344 (3,0)	0,403 (1,0)	0,397 (2,0)
SonyAIBORobotSurface	0,696 (3,0)	0,800 (2,0)	0,900 (1,0)
SonyAIBORobotSurfaceII	0,859 (2,0)	0,889 (1,0)	0,815 (3,0)
StarLightCurves	0,849 (3,0)	0,886 (1,0)	0,865 (2,0)
Strawberry	0,938 (1,0)	0,919 (2,0)	0,905 (3,0)
SwedishLeaf	0,789 (1,0)	0,780 (2,0)	0,778 (3,0)
Symbols	0,899 (1,0)	0,800 (3,0)	0,840 (2,0)
SyntheticControl	0,880 (1,0)	0,803 (2,0)	0,780 (3,0)
ToeSegmentation1	0,680 (1,0)	0,550 (2,0)	0,450 (3,0)
ToeSegmentation2	0,808 (1,0)	0,750 (2,0)	0,639 (3,0)
Trace	0,760 (3,0)	1,000 (1,0)	0,840 (2,0)
TwoLeadECG	0,747 (2,0)	0,739 (3,0)	0,783 (1,0)
TwoPatterns	0,907 (3,0)	0,913 (1,0)	0,909 (2,0)
UWaveGestureLibraryAll	0,948 (1,0)	0,948 (2,0)	0,945 (3,0)
uWaveGestureLibraryX	0,739 (1,0)	0,713 (2,0)	0,708 (3,0)
uWaveGestureLibraryY	0,662 (3,0)	0,683 (2,0)	0,698 (1,0)
uWaveGestureLibraryZ	0,650 (3,0)	0,675 (2,0)	0,690 (1,0)
Wafer	0,995 (2,0)	0,993 (3,0)	0,998 (1,0)
Wine	0,611 (3,0)	0,895 (2,0)	0,982 (1,0)
WordsSynonyms	0,618 (2,0)	0,625 (1,0)	0,614 (3,0)
Worms	0,365 (2,0)	0,377 (1,0)	0,299 (3,0)
WormsTwoClass	0,586 (1,0)	0,571 (2,0)	0,558 (3,0)
Yoga	0,830 (1,0)	0,807 (2,0)	0,780 (3,0)
Médias de ranks	1,935	1,882	2,182

A.3 Parâmetros de distâncias elásticas

A tabela a seguir contém os valores de acurácia observada nos conjuntos do repositório UCR para as distâncias elásticas. Esse experimento foi discutido na Seção 4.7.1.

Conjunto	Acurácia (parâmetros)			
	DTW (δ)	WDTW (g)	LCSS ($\epsilon; \delta$)	EDR (δ)
50Words	0,7648 (16)	0,7868 (0,1)	0,7956 (0,2; 27)	0,7956 (0,3)
Adiac	0,6087 (5)	0,6036 (0,9)	0,6113 (0,05; 1)	0,6164 (0,05)
ArrowHead	0,8000 (0)	0,7943 (0,4)	0,7657 (0,2; 1)	0,7886 (0,2)
Beef	0,6667 (0)	0,7000 (0,5)	0,6000 (0,02; 7)	0,5000 (0,02)
BeetleFly	0,7000 (36)	0,7000 (0)	0,8500 (0,1; 51)	0,7500 (0,03)
BirdChicken	0,7000 (31)	0,7000 (0,1)	0,8000 (0,1; 128)	0,6000 (0,01)
Car	0,7667 (6)	0,8000 (0,4)	0,7667 (0,05; 6)	0,8833 (0,03)
CBF	0,9944 (14)	0,9967 (0)	0,9944 (0,2; 26)	0,9900 (0,4)
ChlorineConcentration	0,6500 (0)	0,6492 (0,9)	0,6583 (0,2; 2)	0,6539 (0,2)
CinCECGtorso	0,9304 (16)	0,9370 (0,1)	0,8188 (0,01; 82)	0,9145 (0,6)
Coffee	1,0000 (0)	1,0000 (0,3)	0,9286 (0,1; 9)	1,0000 (0,2)
Computers	0,6200 (94)	0,7000 (0)	0,6320 (0,05; 7)	0,6440 (0,06)
CricketX	0,7718 (30)	0,7538 (0)	0,6872 (0,2; 45)	0,7051 (0,3)
CricketY	0,7615 (51)	0,7513 (0,1)	0,7821 (0,5; 30)	0,7615 (0,5)
CricketZ	0,7462 (15)	0,7538 (0,1)	0,7410 (0,5; 30)	0,7333 (0,3)
DiatomSizeReduction	0,9346 (0)	0,9673 (0)	0,9118 (0,01; 1)	0,9510 (0,05)
DistalPhalanxOutlineAgeGroup	0,7725 (1)	0,7800 (0,7)	0,7275 (0,05; 7)	0,7350 (0,05)
DistalPhalanxOutlineCorrect	0,7683 (2)	0,7633 (0,3)	0,7600 (0,1; 2)	0,7133 (0,02)
DistalPhalanxTW	0,7275 (0)	0,7325 (0,6)	0,7350 (0,1; 1)	0,7275 (0,08)

Earthquakes	0,7422	(113)	0,7422	(0)	0,7422	(0,5; 3)	0,7826	(0,07)
ECG200	0,8800	(0)	0,8700	(0,8)	0,8700	(0,1; 1)	0,9000	(0,4)
ECG5000	0,9251	(1)	0,9253	(0,6)	0,9300	(0,1; 35)	0,9284	(0,05)
ECGFiveDays	0,7967	(0)	0,8049	(0,5)	0,7294	(0,02; 27)	0,9489	(0,6)
ElectricDevices	0,5508	(13)	0,6172	(0,1)	0,6199	(0,5; 6)	0,6217	(0,07)
FaceAll	0,8083	(4)	0,7905	(0,1)	0,8059	(0,5; 5)	0,8077	(0,5)
FaceFour	0,8864	(7)	0,8750	(0,2)	0,8977	(0,05; 4)	0,9205	(0,04)
FacesUCR	0,9122	(16)	0,9190	(0,1)	0,9561	(0,5; 13)	0,9595	(0,4)
Fish	0,8457	(19)	0,8457	(0,5)	0,8914	(0,05; 46)	0,9257	(0,03)
FordA	0,6590	(0)	0,6457	(0,8)	0,6893	(0,5; 3)	0,6759	(0,6)
FordB	0,5861	(5)	0,5960	(0,1)	0,6557	(0,5; 75)	0,6650	(0,6)
GunPoint	0,9133	(0)	0,9667	(0,2)	0,9600	(0,2; 1)	0,9800	(0,2)
Ham	0,6000	(0)	0,5714	(0,4)	0,5810	(0,2; 3)	0,6286	(0,3)
HandOutlines	0,8030	(27)	0,8080	(0,2)	0,8080	(0,05; 271)	0,8050	(0,06)
Haptics	0,4123	(22)	0,3571	(0,9)	0,3312	(0,5; 5)	0,3701	(0,4)
Herring	0,5312	(26)	0,5312	(0)	0,5469	(0,05; 26)	0,5000	(0,01)
InlineSkate	0,3873	(263)	0,3836	(0)	0,4327	(0,2; 94)	0,4236	(0,07)
InsectWingbeatSound	0,5783	(5)	0,5793	(0,7)	0,5535	(0,05; 6)	0,5131	(0,05)
ItalyPowerDemand	0,9553	(0)	0,9504	(0)	0,8737	(0,1; 1)	0,9281	(0,2)
LargeKitchenAppliances	0,7947	(677)	0,7947	(0)	0,5707	(0,5; 36)	0,5387	(0,06)
Lightning2	0,8689	(38)	0,8689	(0)	0,8361	(0,1; 2)	0,8361	(0,1)
Lightning7	0,7123	(16)	0,7945	(0,1)	0,5890	(0,5; 8)	0,6575	(0,4)
MALLAT	0,9143	(0)	0,9373	(0,2)	0,8810	(0,5; 3)	0,8802	(0,3)
Meat	0,9333	(0)	0,9333	(0)	0,9000	(0,01; 1)	0,9167	(0,01)
MedicalImages	0,7474	(20)	0,7382	(0,2)	0,6737	(0,2; 25)	0,6553	(0,1)
MiddlePhalanxOutlineAgeGroup	0,7475	(4)	0,7500	(0)	0,2700	(0,5; 1)	0,7275	(0,4)
MiddlePhalanxOutlineCorrect	0,6817	(1)	0,7083	(0,6)	0,7283	(0,1; 1)	0,7483	(0,1)
MiddlePhalanxTW	0,5840	(2)	0,5915	(0,1)	0,6040	(0,1; 1)	0,5815	(0,08)
MoteStrain	0,8658	(1)	0,8570	(0,3)	0,8299	(0,01; 4)	0,8323	(0,01)
NonInvasiveFetalECG1	0,8107	(8)	0,8102	(0,3)	0,8158	(0,2; 5)	0,8153	(0,2)
NonInvasiveFetalECG2	0,8748	(8)	0,8865	(0,6)	0,8763	(0,2; 38)	0,8794	(0,2)
OliveOil	0,8667	(0)	0,8333	(0,1)	0,8667	(0,01; 1)	0,8667	(0,01)
OSULeaf	0,6116	(30)	0,5992	(0,1)	0,8058	(0,1; 128)	0,7562	(0,2)
PhalangesOutlinesCorrect	0,7611	(0)	0,7424	(0,6)	0,7156	(0,1; 1)	0,7156	(0,07)
Phoneme	0,2268	(143)	0,1614	(0,1)	0,2695	(0,5; 102)	0,2352	(0,4)
Plane	1,0000	(9)	1,0000	(0)	1,0000	(0,05; 7)	1,0000	(0,04)
ProximalPhalanxOutlineAgeGroup	0,7854	(0)	0,8049	(0)	0,7951	(0,02; 2)	0,7805	(0,09)
ProximalPhalanxOutlineCorrect	0,7904	(1)	0,7835	(0,2)	0,7698	(0,1; 3)	0,7766	(0,07)
ProximalPhalanxTW	0,7375	(5)	0,7425	(0,1)	0,7350	(0,1; 2)	0,7275	(0,08)
RefrigerationDevices	0,4400	(58)	0,4160	(0,1)	0,5467	(0,1; 108)	0,4880	(0,2)
ScreenType	0,4107	(122)	0,3787	(0,1)	0,4720	(0,1; 216)	0,4293	(0,07)
ShapeletSim	0,7000	(15)	0,7444	(0,1)	0,9111	(0,5; 50)	0,8611	(0,5)
ShapesAll	0,8017	(20)	0,8083	(0,1)	0,8333	(0,05; 51)	0,8700	(0,05)
SmallKitchenAppliances	0,6720	(108)	0,5867	(0,1)	0,6187	(0,01; 1)	0,5173	(0,01)
SonyAIBORobotSurface	0,6955	(0)	0,7421	(0,9)	0,6722	(0,2; 1)	0,7304	(0,5)
SonyAIBORobotSurfaceII	0,8594	(0)	0,8311	(0)	0,7922	(0,05; 4)	0,8279	(0,09)
StarLightCurves	0,8488	(164)		(0)	0,8813	(0,2; 102)	0,8813	(0,1)
Strawberry	0,9380	(0)	0,9380	(0,8)	0,9315	(0,02; 1)	0,9217	(0,02)
SwedishLeaf	0,8464	(3)	0,8736	(0,4)	0,8864	(0,2; 9)	0,9024	(0,2)
Symbols	0,9377	(32)	0,9497	(0)	0,9578	(0,2; 60)	0,9568	(0,2)
SyntheticControl	0,9833	(4)	0,9933	(0)	0,9567	(0,5; 4)	0,9533	(0,6)
ToeSegmentation1	0,7500	(22)	0,7719	(0)	0,8114	(0,05; 55)	0,7939	(0,1)
ToeSegmentation2	0,9077	(17)	0,8923	(0,1)	0,8769	(0,02; 34)	0,9308	(0,4)
Trace	0,9900	(8)	1,0000	(0)	0,9700	(0,2; 8)	0,9100	(0,06)
TwoLeadECG	0,8683	(4)	0,9043	(0)	0,9034	(0,1; 5)	0,9447	(0,08)
TwoPatterns	0,9972	(5)	1,0000	(0)	0,9998	(0,2; 13)	0,9985	(0,2)
UWaveGestureLibraryAll	0,9657	(38)	0,9609	(0,1)	0,9604	(0,2; 47)	0,9606	(0,3)

uWaveGestureLibraryX	0,7733	(13)	0,7753	(0,1)	0,7605	(0,5; 63)	0,7596	(0,5)
uWaveGestureLibraryY	0,6991	(13)	0,6946	(0,1)	0,6756	(0,05; 16)	0,6879	(0,4)
uWaveGestureLibraryZ	0,6778	(19)	0,6784	(0,1)	0,6776	(0,5; 32)	0,6840	(0,3)
Wafer	0,9955	(2)	0,9971	(0,5)	0,9906	(0,2; 1)	0,9895	(0,2)
Wine	0,6111	(0)	0,5741	(0)	0,5185	(0,01; 1)	0,5000	(0,01)
WordsSynonyms	0,7476	(22)	0,7508	(0,1)	0,7555	(0,5; 41)	0,7445	(0,3)
Worms	0,4144	(27)	0,4309	(0,1)	0,4751	(0,2; 225)	0,4751	(0,05)
WormsTwoClass	0,5856	(81)	0,6243	(0,1)	0,6519	(0,5; 5)	0,6077	(0,05)
Yoga	0,8447	(9)	0,8467	(0,2)	0,8597	(0,05; 64)	0,8687	(0,03)

A.4 Classificador 1-NN dinâmico

Na tabela a seguir apresenta-se os valores de acurácia e as melhores distâncias selecionadas pelo 1-NN dinâmico com e sem distâncias elásticas. Esse experimento é discutido na Seção 4.7.2.2.

Conjunto	Classificadores (distâncias selecionadas)			
	1-NN dinâmico	1-NN elástico	1-NN	1-NN DTW
50Words	0,7011 (χ^2)	0,7956 (EDR)	0,6308	0,7648
Adiac	0,6164 (L_∞)	0,6164 (L_∞)	0,6113	0,6087
ArrowHead	0,8000 (Euclidiana)	0,8000 (DTW)	0,8000	0,8000
Beef	0,6667 (Euclidiana)	0,6667 (DTW)	0,6667	0,6667
BeetleFly	0,6500 (Lorentzian)	0,8500 (LCSS)	0,7500	0,7000
BirdChicken	0,5500 (Pearson)	0,8000 (LCSS)	0,5500	0,7000
Car	0,7167 (L_1/L_∞)	0,7667 (LCSS)	0,7333	0,7667
CBF	0,9522 (χ^2)	0,9944 (DTW)	0,8522	0,9944
ChlorineConcentration	0,7125 (Bhatt.)	0,7125 (Bhatt.)	0,6500	0,6500
CinCECGtorso	0,9326 (Lorentzian)	0,9304 (DTW)	0,8971	0,9304
Coffee	0,9643 (L_1/L_∞)	1,0000 (DTW)	1,0000	1,0000
Computers	0,5840 (KL)	0,6200 (DTW)	0,5760	0,6200
CricketX	0,6308 (L_1/L_∞)	0,7718 (DTW)	0,5769	0,7718
CricketY	0,6564 (Lorentzian)	0,7821 (LCSS)	0,5667	0,7615
CricketZ	0,6590 (L_1)	0,7538 (WDTW)	0,5872	0,7462
DiatomSizeReduction	0,9281 (L_1/L_∞)	0,9346 (DTW)	0,9346	0,9346
DistalPhalanxOutlineAgeGroup	0,7900 (L_1/L_∞)	0,7800 (WDTW)	0,7825	0,7725
DistalPhalanxOutlineCorrect	0,7483 (Bhatt.)	0,7633 (WDTW)	0,7517	0,7683
DistalPhalanxTW	0,7275 (Pearson)	0,7325 (WDTW)	0,7275	0,7275
Earthquakes	0,8137 (Jeffrey)	0,7422 (LCSS)	0,6739	0,7422
ECG200	0,8900 (Lorentzian)	0,9000 (EDR)	0,8800	0,8800
ECG5000	0,9249 (L_1/L_∞)	0,9300 (LCSS)	0,9249	0,9251
ECGFiveDays	0,7967 (Euclidiana)	0,7294 (LCSS)	0,7967	0,7967
ElectricDevices	0,5715 (Lorentzian)	0,6172 (WDTW)	0,5508	0,5508
FaceAll	0,7325 (Canberra)	0,8077 (EDR)	0,7136	0,8083
FaceFour	0,8636 (Canberra)	0,8977 (LCSS)	0,7841	0,8864
FacesUCR	0,8210 (Canberra)	0,9561 (LCSS)	0,7693	0,9122
Fish	0,7943 (L_1)	0,9257 (EDR)	0,7829	0,8457
FordA	0,6487 (Lorentzian)	0,6893 (LCSS)	0,6590	0,6590
FordB	0,5756 (L_1/L_∞)	0,6650 (EDR)	0,5578	0,5861
GunPoint	0,9133 (Euclidiana)	0,9600 (LCSS)	0,9133	0,9133
Ham	0,5048 (L_1/L_∞)	0,5048 (L_1/L_∞)	0,6000	0,6000
HandOutlines	0,8170 (L_∞)	0,8030 (DTW)	0,8010	0,8030
Haptics	0,3896 (L_∞)	0,3701 (EDR)	0,3701	0,4123
Herring	0,5938 (Bhatt.)	0,5000 (EDR)	0,5156	0,5312
InlineSkate	0,3273 (χ^2)	0,4327 (LCSS)	0,3418	0,3873

InsectWingbeatSound	0,5909	(KL)	0,5535	(LCSS)	0,5616	0,5783
ItalyPowerDemand	0,9534	(Jeffrey)	0,9534	(Jeffrey)	0,9553	0,9553
LargeKitchenAppliances	0,5627	(L_1/L_∞)	0,7947	(DTW)	0,4933	0,7947
Lightning2	0,7705	(Pearson)	0,8689	(DTW)	0,7541	0,8689
Lightning7	0,6712	(L_1/L_∞)	0,7123	(DTW)	0,5753	0,7123
MALLAT	0,9143	(Euclidiana)	0,9143	(DTW)	0,9143	0,9143
Meat	0,9333	(L_1/L_∞)	0,9333	(DTW)	0,9333	0,9333
MedicalImages	0,7145	(KL)	0,7474	(DTW)	0,6842	0,7474
MiddlePhalanxOutlineAgeGroup	0,7275	(Canberra)	0,7275	(EDR)	0,7400	0,7475
MiddlePhalanxOutlineCorrect	0,7533	(Euclidiana)	0,6817	(DTW)	0,7533	0,6817
MiddlePhalanxTW	0,6015	(Canberra)	0,5915	(WDTW)	0,5614	0,5840
MoteStrain	0,8778	(Canberra)	0,8299	(LCSS)	0,8786	0,8658
NonInvasiveFetalECG1	0,8377	(KL)	0,8102	(WDTW)	0,8290	0,8107
NonInvasiveFetalECG2	0,8799	(Euclidiana)	0,8799	(Euclidiana)	0,8799	0,8748
OliveOil	0,8667	(Bhatt.)	0,8667	(Bhatt.)	0,8667	0,8667
OSULeaf	0,5496	(χ^2)	0,8058	(LCSS)	0,5207	0,6116
PhalangesOutlinesCorrect	0,7611	(Euclidiana)	0,7611	(DTW)	0,7611	0,7611
Phoneme	0,1081	(L_1/L_∞)	0,2352	(EDR)	0,1092	0,2268
Plane	0,9619	(Jeffrey)	1,0000	(DTW)	0,9619	1,0000
ProximalPhalanxOutlineAgeGroup	0,7951	(Pearson)	0,7805	(EDR)	0,7854	0,7854
ProximalPhalanxOutlineCorrect	0,8110	(L_∞)	0,8110	(L_∞)	0,8076	0,7904
ProximalPhalanxTW	0,7050	(χ^2)	0,7275	(EDR)	0,7075	0,7375
RefrigerationDevices	0,4507	(Sørensen)	0,4880	(EDR)	0,3947	0,4400
ScreenType	0,3600	(L_1)	0,4107	(DTW)	0,3600	0,4107
ShapeletSim	0,4778	(L_1/L_∞)	0,9111	(LCSS)	0,5389	0,7000
ShapesAll	0,7583	(χ^2)	0,8700	(EDR)	0,7517	0,8017
SmallKitchenAppliances	0,4880	(Lorentzian)	0,6720	(DTW)	0,3440	0,6720
SonyAIBORobotSurface	0,6922	(L_1/L_∞)	0,6955	(DTW)	0,6955	0,6955
SonyAIBORobotSurfaceII	0,8510	(Canberra)	0,8279	(EDR)	0,8594	0,8594
StarLightCurves	0,8830	(Jeffrey)		(WDTW)	0,8488	0,8488
Strawberry	0,9478	(Canberra)	0,9217	(EDR)	0,9380	0,9380
SwedishLeaf	0,7888	(L_1)	0,9024	(EDR)	0,7888	0,8464
Symbols	0,8814	(Bhatt.)	0,9578	(LCSS)	0,8995	0,9377
SyntheticControl	0,8800	(L_1/L_∞)	0,9833	(DTW)	0,8800	0,9833
ToeSegmentation1	0,7325	(Lorentzian)	0,8114	(LCSS)	0,6798	0,7500
ToeSegmentation2	0,8462	(Canberra)	0,8769	(LCSS)	0,8077	0,9077
Trace	0,9900	(χ^2)	1,0000	(WDTW)	0,7600	0,9900
TwoLeadECG	0,7419	(Canberra)	0,9034	(LCSS)	0,7471	0,8683
TwoPatterns	0,9647	(Sørensen)	0,9998	(LCSS)	0,9067	0,9972
UWaveGestureLibraryAll	0,9545	(L_1)	0,9657	(DTW)	0,9481	0,9657
uWaveGestureLibraryX	0,7393	(Euclidiana)	0,7605	(LCSS)	0,7393	0,7733
uWaveGestureLibraryY	0,6781	(χ^2)	0,6991	(DTW)	0,6616	0,6991
uWaveGestureLibraryZ	0,6633	(Bhatt.)	0,6776	(LCSS)	0,6496	0,6778
Wafer	0,9972	(χ^2)	0,9955	(DTW)	0,9955	0,9955
Wine	0,6481	(L_1/L_∞)	0,6111	(DTW)	0,6111	0,6111
WordsSynonyms	0,6599	(χ^2)	0,7555	(LCSS)	0,6176	0,7476
Worms	0,3481	(χ^2)	0,4751	(LCSS)	0,3646	0,4144
WormsTwoClass	0,6077	(L_1/L_∞)	0,6519	(LCSS)	0,5856	0,5856
Yoga	0,8130	(KL)	0,8687	(EDR)	0,8303	0,8447

A.5 Gráficos de distância críticas para os *ensembles*

Os gráficos a seguir indicam a distância crítica para os grupos de métodos de composição de votos discutidos na Seção 4.7.2.2.

Figura 65 – Gráfico CD para o *ensemble* majoritário ponderado por acurácia. Figura 66 – Gráfico CD para o *ensemble* de composição de *rankings* de probabilidade a posteriori. Figura 67 – Gráfico CD para o *ensemble* majoritário ponderado por probabilidade a posteriori. Figura 68 – Gráfico CD para o *ensemble* baseado em *simple rank*.

A.6 Teste do *ensemble* dinâmico

Na tabela a seguir, valores de acurácia para o *ensemble* dinâmico e com composição de votos ponderados pela acurácia. Para comparação, o 1-NN DTW e o 1-NN com distância euclidiana. Note que os valores de acurácia apresentados para o 1-NN não correspondem mais, para alguns conjuntos de dados, aos valores que serão obtidos caso os experimentos sejam realizados com os novos conjuntos do repositório UCR. A discussão desses experimentos foi realizada na Seção 6.5.1.

Conjunto	Ensemble ou distância no domínio temporal (<i>rank</i> s)			
	Dinâmico	Acurácia	DTW	Euclidiana
50Words	0,7780 (1,0)	0,7516 (3,0)	0,7648 (2,0)	0,6308 (4,0)
Adiac	0,7138 (2,0)	0,7263 (1,0)	0,6087 (4,0)	0,6113 (3,0)
Beef	0,5333 (3,5)	0,5333 (3,5)	0,5333 (1,5)	0,5333 (1,5)
CBF	0,9933 (2,5)	0,9933 (2,5)	0,9944 (1,0)	0,8522 (4,0)
ChlorineConcentration	0,6523 (2,0)	0,6526 (1,0)	0,6500 (3,5)	0,6500 (3,5)

CinCEGTorso	0,9246 (2,5)	0,9246 (2,5)	0,9304 (1,0)	0,8971 (4,0)
Coffee	0,8214 (2,5)	0,8214 (2,5)	0,8214 (1,0)	0,7500 (4,0)
CricketX	0,7769 (1,5)	0,7769 (1,5)	0,7641 (3,0)	0,5744 (4,0)
CricketY	0,8077 (1,5)	0,8077 (1,5)	0,8026 (3,0)	0,6436 (4,0)
CricketZ	0,8282 (2,0)	0,8359 (1,0)	0,8205 (3,0)	0,6205 (4,0)
DiatomSizeReduction	0,9428 (1,0)	0,9425 (2,0)	0,9346 (3,5)	0,9346 (3,5)
ECG200	0,8940 (1,0)	0,8800 (3,0)	0,8800 (3,0)	0,8800 (3,0)
ECGFiveDays	0,9384 (1,5)	0,9384 (1,5)	0,7967 (3,5)	0,7967 (3,5)
FaceAll	0,8160 (2,0)	0,8178 (1,0)	0,8083 (3,0)	0,7136 (4,0)
FaceFour	0,8523 (2,0)	0,8409 (3,0)	0,8864 (1,0)	0,7841 (4,0)
FacesUCR	0,9063 (2,5)	0,9063 (2,5)	0,9122 (1,0)	0,7693 (4,0)
Fish	0,8891 (1,0)	0,8743 (2,0)	0,8400 (3,0)	0,7829 (4,0)
GunPoint	0,9667 (1,0)	0,9200 (2,0)	0,9133 (3,5)	0,9133 (3,5)
Haptics	0,4286 (2,0)	0,4351 (1,0)	0,4123 (3,0)	0,3701 (4,0)
InlineSkate	0,3909 (1,5)	0,3909 (1,5)	0,3873 (3,0)	0,3418 (4,0)
ItalyPowerDemand	0,9553 (2,0)	0,9563 (1,0)	0,9553 (3,5)	0,9553 (3,5)
Lightning2	0,8689 (1,5)	0,8689 (1,5)	0,8689 (3,0)	0,7541 (4,0)
Lightning7	0,7534 (1,5)	0,7534 (1,5)	0,7123 (3,0)	0,5753 (4,0)
MALLAT	0,9235 (1,0)	0,9177 (2,0)	0,9143 (3,5)	0,9143 (3,5)
MedicalImages	0,7487 (1,0)	0,7368 (3,0)	0,7474 (2,0)	0,6842 (4,0)
MoteStrain	0,8722 (3,0)	0,8786 (1,0)	0,8658 (4,0)	0,8786 (2,0)
NonInvasiveFetalECG1	0,8479 (2,0)	0,8539 (1,0)	0,8107 (4,0)	0,8290 (3,0)
NonInvasiveFetalECG2	0,9159 (1,0)	0,9120 (2,0)	0,8748 (4,0)	0,8799 (3,0)
OliveOil	0,8333 (3,5)	0,8333 (3,5)	0,8333 (2,0)	0,8667 (1,0)
OSULeaf	0,6322 (1,5)	0,6322 (1,5)	0,6157 (3,0)	0,5165 (4,0)
SonyAIBORobotSurface	0,7005 (1,5)	0,7005 (1,5)	0,6955 (3,5)	0,6955 (3,5)
SonyAIBORobotSurfacell	0,8399 (4,0)	0,8625 (1,0)	0,8594 (2,5)	0,8594 (2,5)
StarLightCurves	0,9326 (1,5)	0,9326 (1,5)	0,9053 (3,0)	0,8488 (4,0)
SwedishLeaf	0,9024 (1,0)	0,8976 (2,0)	0,8464 (3,0)	0,7888 (4,0)
Symbols	0,9397 (2,0)	0,9473 (1,0)	0,9377 (3,0)	0,8995 (4,0)
SyntheticControl	0,9933 (1,5)	0,9933 (1,5)	0,9833 (3,0)	0,8800 (4,0)
Trace	0,9800 (2,0)	0,9400 (3,0)	0,9900 (1,0)	0,7600 (4,0)
TwoLeadECG	0,8745 (1,5)	0,8745 (1,5)	0,8683 (3,0)	0,7471 (4,0)
TwoPatterns	0,9980 (1,5)	0,9980 (1,5)	0,9972 (3,0)	0,9067 (4,0)
uWaveGestureLibraryX	0,7753 (1,5)	0,7753 (1,5)	0,7733 (3,0)	0,7393 (4,0)
uWaveGestureLibraryY	0,6926 (2,5)	0,6926 (2,5)	0,6991 (1,0)	0,6616 (4,0)
uWaveGestureLibraryZ	0,6882 (1,5)	0,6882 (1,5)	0,6778 (3,0)	0,6496 (4,0)
Wafer	0,9972 (1,5)	0,9972 (1,5)	0,9955 (3,5)	0,9955 (3,5)
WordsSynonyms	0,7367 (2,5)	0,7367 (2,5)	0,7476 (1,0)	0,6176 (4,0)
Yoga	0,8633 (1,0)	0,8593 (2,0)	0,8447 (3,0)	0,8303 (4,0)
Ranks médios	1,81	1,86	2,71	3,62

A.7 “Atualização” do *ensemble* de acurácias

O *ensemble* de acurácias foi avaliado sobre os novos conjuntos do repositório UCR, conforme discutido na Seção 6.5.2. Na tabela a seguir são apresentados os valores de acurácia para o *ensemble* de acurácias com seleção de distância (sel.), o *ensemble* de acurácias utilizando distância euclidiana (Euc) e o classificador 1-NN DTW.

Conjunto	Classificador (rank)		
	<i>Ensemble de Acurácias (dinâm.)</i>	<i>Ensemble de Acurácias (euc.)</i>	1-NN DTW
50Words	0,7692 (1,0)	0,7516 (3,0)	0,7648 (2,0)

Adiac	0,7391	(1,0)	0,6982	(2,0)	0,6087	(3,0)
ArrowHead	0,8229	(1,0)	0,8000	(2,5)	0,8000	(2,5)
Beef	0,7000	(1,0)	0,6667	(2,5)	0,6667	(2,5)
BeetleFly	0,7000	(2,0)	0,7000	(2,0)	0,7000	(2,0)
BirdChicken	0,7500	(1,0)	0,7000	(2,5)	0,7000	(2,5)
Car	0,7667	(2,0)	0,7667	(2,0)	0,7667	(2,0)
CBF	0,9922	(2,0)	0,9867	(3,0)	0,9944	(1,0)
ChlorineConcentration	0,6937	(1,0)	0,6503	(2,0)	0,6500	(3,0)
CinCECGtorso	0,9688	(1,0)	0,9232	(3,0)	0,9304	(2,0)
Coffee	0,9643	(3,0)	1,0000	(1,5)	1,0000	(1,5)
Computers	0,6880	(1,0)	0,6600	(2,0)	0,6200	(3,0)
CricketX	0,7615	(2,0)	0,7538	(3,0)	0,7718	(1,0)
CricketY	0,7564	(2,0)	0,7231	(3,0)	0,7615	(1,0)
CricketZ	0,7641	(1,5)	0,7641	(1,5)	0,7462	(3,0)
DiatomSizeReduction	0,9477	(1,5)	0,9477	(1,5)	0,9346	(3,0)
DistalPhalanxOutlineAgeGroup	0,8050	(1,0)	0,7950	(2,0)	0,7725	(3,0)
DistalPhalanxOutlineCorrect	0,8000	(1,0)	0,7883	(2,0)	0,7683	(3,0)
DistalPhalanxTW	0,7475	(1,0)	0,7425	(2,0)	0,7275	(3,0)
Earthquakes	0,8199	(1,0)	0,7795	(2,0)	0,7422	(3,0)
ECG200	0,9000	(1,0)	0,8800	(2,5)	0,8800	(2,5)
ECG5000	0,9384	(1,0)	0,9367	(2,0)	0,9251	(3,0)
ECGFiveDays	0,9582	(1,0)	0,9175	(2,0)	0,7967	(3,0)
ElectricDevices	0,6620	(1,0)	0,6404	(2,0)	0,5508	(3,0)
FaceAll	0,8089	(2,0)	0,8260	(1,0)	0,8083	(3,0)
FaceFour	0,8636	(2,0)	0,8523	(3,0)	0,8864	(1,0)
FacesUCR	0,9180	(1,0)	0,9132	(2,0)	0,9122	(3,0)
Fish	0,9257	(1,0)	0,8686	(2,0)	0,8457	(3,0)
FordA	0,8770	(1,0)	0,7881	(2,0)	0,6590	(3,0)
FordB	0,8171	(1,0)	0,7880	(2,0)	0,5861	(3,0)
GunPoint	0,9667	(1,0)	0,9200	(2,0)	0,9133	(3,0)
Ham	0,5333	(3,0)	0,6000	(1,5)	0,6000	(1,5)
HandOutlines	0,8280	(1,0)	0,8180	(2,0)	0,8030	(3,0)
Haptics	0,4610	(1,0)	0,4253	(2,0)	0,4123	(3,0)
Herring	0,5938	(2,0)	0,6094	(1,0)	0,5312	(3,0)
InlineSkate	0,4345	(1,0)	0,3909	(2,0)	0,3873	(3,0)
InsectWingbeatSound	0,6045	(1,0)	0,5889	(2,0)	0,5783	(3,0)
ItalyPowerDemand	0,9572	(1,0)	0,9553	(2,5)	0,9553	(2,5)
LargeKitchenAppliances	0,8027	(1,0)	0,8000	(2,0)	0,7947	(3,0)
Lightning2	0,8525	(3,0)	0,8852	(1,0)	0,8689	(2,0)
Lightning7	0,7260	(2,0)	0,7534	(1,0)	0,7123	(3,0)
MALLAT	0,9224	(1,0)	0,9143	(2,5)	0,9143	(2,5)
Meat	0,9333	(2,0)	0,9333	(2,0)	0,9333	(2,0)
MedicalImages	0,7526	(1,0)	0,7500	(2,0)	0,7474	(3,0)
MiddlePhalanxOutlineAgeGroup	0,7650	(1,0)	0,7500	(2,0)	0,7475	(3,0)
MiddlePhalanxOutlineCorrect	0,7517	(1,0)	0,7467	(2,0)	0,6817	(3,0)
MiddlePhalanxTW	0,5990	(1,0)	0,5840	(2,5)	0,5840	(2,5)
MoteStrain	0,9034	(1,0)	0,8786	(2,0)	0,8658	(3,0)
NonInvasiveFetalECG1	0,8590	(1,0)	0,8453	(2,0)	0,8107	(3,0)
NonInvasiveFetalECG2	0,9008	(2,0)	0,9028	(1,0)	0,8748	(3,0)
OliveOil	0,8333	(3,0)	0,8667	(1,5)	0,8667	(1,5)
OSULeaf	0,6777	(1,0)	0,6198	(2,0)	0,6116	(3,0)
PhalangesOutlinesCorrect	0,7797	(1,0)	0,7762	(2,0)	0,7611	(3,0)
Phoneme	0,2690	(1,0)	0,2342	(2,0)	0,2268	(3,0)
Plane	1,0000	(1,0)	0,9714	(3,0)	1,0000	(2,0)
ProximalPhalanxOutlineAgeGroup	0,8098	(1,0)	0,7902	(2,0)	0,7854	(3,0)
ProximalPhalanxOutlineCorrect	0,8316	(1,0)	0,8282	(2,0)	0,7904	(3,0)
ProximalPhalanxTW	0,7575	(1,0)	0,7450	(2,0)	0,7375	(3,0)

RefrigerationDevices	0,4640	(1,5)	0,4640	(1,5)	0,4400	(3,0)
ScreenType	0,4027	(3,0)	0,4160	(1,0)	0,4107	(2,0)
ShapeletSim	0,7667	(2,0)	0,7944	(1,0)	0,7000	(3,0)
ShapesAll	0,8350	(1,0)	0,8117	(2,0)	0,8017	(3,0)
SmallKitchenAppliances	0,7200	(1,0)	0,5973	(3,0)	0,6720	(2,0)
SonyAIBORobotSurface	0,7105	(1,0)	0,7055	(2,0)	0,6955	(3,0)
SonyAIBORobotSurfacell	0,8762	(1,0)	0,8604	(2,0)	0,8594	(3,0)
StarLightCurves	0,9671	(1,0)	0,9336	(2,0)	0,8488	(3,0)
Strawberry	0,9527	(1,0)	0,9380	(2,5)	0,9380	(2,5)
SwedishLeaf	0,9104	(1,0)	0,8976	(2,0)	0,8464	(3,0)
Symbols	0,9417	(2,0)	0,9457	(1,0)	0,9377	(3,0)
SyntheticControl	0,9900	(1,5)	0,9900	(1,5)	0,9833	(3,0)
ToeSegmentation1	0,7982	(1,0)	0,7851	(2,0)	0,7500	(3,0)
ToeSegmentation2	0,9077	(1,0)	0,8846	(3,0)	0,9077	(2,0)
Trace	0,9900	(1,0)	0,9400	(3,0)	0,9900	(2,0)
TwoLeadECG	0,8788	(2,0)	0,8850	(1,0)	0,8683	(3,0)
TwoPatterns	0,9980	(1,0)	0,9968	(3,0)	0,9972	(2,0)
UWaveGestureLibraryAll	0,9626	(2,0)	0,9595	(3,0)	0,9657	(1,0)
uWaveGestureLibraryX	0,7875	(1,0)	0,7725	(3,0)	0,7733	(2,0)
uWaveGestureLibraryY	0,7105	(1,0)	0,6896	(3,0)	0,6991	(2,0)
uWaveGestureLibraryZ	0,6991	(1,0)	0,6865	(2,0)	0,6778	(3,0)
Wafer	0,9989	(1,0)	0,9969	(2,0)	0,9955	(3,0)
Wine	0,5926	(2,0)	0,5741	(3,0)	0,6111	(1,0)
WordsSynonyms	0,7414	(2,0)	0,7335	(3,0)	0,7476	(1,0)
Worms	0,4530	(2,0)	0,4586	(1,0)	0,4144	(3,0)
WormsTwoClass	0,6796	(1,0)	0,6630	(2,0)	0,5856	(3,0)
Yoga	0,8617	(1,0)	0,8580	(2,0)	0,8447	(3,0)
<i>Ranks médios</i>	1,35		2,08		2,57	

A.8 Outros resultados

As tabelas nas próximas páginas dizem respeito a resultados experimentais apresentados nos Capítulos 5 e 6.

A tabela “**Representações individuais**” contém os resultados completos do experimento de diversidade de representações, utilizando distância euclidiana, distâncias discutido na Seção 5.8.2, bem como os *ranks* obtidos através do teste de Friedman.

A tabela “**Teste de Friedman para o primeiro experimento de ensembles**” contém valores de acurácia para os *ensembles* discutidos na primeira parte da Seção 6.5.1, bem como para o 1-NN DTW, acompanhado dos *ranks* obtidos para a realização do teste de Friedman. Na tabela, P.P. significa “probabilidade a posteriori” e “majoritário” refere-se à acurácia simples.

“Representações individuais”

Conjunto	Acurácias nos exemplos de teste (<i>rank</i> s)						
	Tempo	PS	DWT	PAA	SAX	PCA	ACF
50Words	0,6308 (5,0)	0,4637 (7,0)	0,6396 (3,0)	0,6527 (2,0)	0,6879 (1,0)	0,6374 (4,0)	0,5714 (6,0)
Adiac	0,6113 (4,0)	0,7315 (1,0)	0,6189 (2,0)	0,6036 (5,0)	0,4425 (7,0)	0,6138 (3,0)	0,5550 (6,0)
ArrowHead	0,8000 (1,0)	0,7657 (5,0)	0,8000 (2,0)	0,7943 (4,0)	0,7543 (7,0)	0,8000 (3,0)	0,7657 (6,0)
Beef	0,6667 (2,0)	0,7000 (1,0)	0,6667 (3,5)	0,6667 (3,5)	0,6333 (6,0)	0,6667 (5,0)	0,6333 (7,0)
BeetleFly	0,7500 (3,5)	0,7500 (3,5)	0,7000 (6,0)	0,8000 (2,0)	0,8500 (1,0)	0,7500 (5,0)	0,7000 (7,0)
BirdChicken	0,5500 (5,0)	0,8000 (1,0)	0,6000 (4,0)	0,5500 (6,0)	0,7000 (2,0)	0,5500 (7,0)	0,7000 (3,0)
Car	0,7333 (1,0)	0,7167 (3,0)	0,6333 (7,0)	0,7167 (4,5)	0,7167 (4,5)	0,7333 (2,0)	0,7167 (6,0)
CBF	0,8522 (4,0)	0,6156 (7,0)	0,8400 (5,0)	0,9433 (1,0)	0,9011 (2,0)	0,8533 (3,0)	0,7644 (6,0)
ChlorineConcentration	0,6500 (3,0)	0,6557 (1,0)	0,6505 (2,0)	0,5701 (7,0)	0,5992 (6,0)	0,6479 (4,0)	0,6083 (5,0)
CinCECGtorso	0,8971 (2,0)	0,8181 (6,0)	0,9043 (1,0)	0,8957 (4,0)	0,8935 (5,0)	0,8971 (3,0)	0,8080 (7,0)
Coffee	1,0000 (1,5)	1,0000 (1,5)	0,9643 (5,0)	1,0000 (3,0)	0,8929 (7,0)	1,0000 (4,0)	0,9286 (6,0)
Computers	0,5760 (4,0)	0,6080 (1,0)	0,5840 (2,0)	0,5240 (6,0)	0,5080 (7,0)	0,5800 (3,0)	0,5600 (5,0)
CricketX	0,5769 (6,0)	0,5923 (5,0)	0,6051 (2,0)	0,6026 (3,0)	0,6282 (1,0)	0,5974 (4,0)	0,5744 (7,0)
CricketY	0,5667 (3,0)	0,5410 (5,0)	0,5590 (4,0)	0,5692 (2,0)	0,5359 (6,0)	0,5769 (1,0)	0,5154 (7,0)
CricketZ	0,5872 (5,0)	0,6359 (2,0)	0,5846 (6,0)	0,6282 (3,0)	0,6103 (4,0)	0,6385 (1,0)	0,5846 (7,0)
DiatomSizeReduction	0,9346 (3,5)	0,9346 (3,5)	0,9379 (1,5)	0,9379 (1,5)	0,9020 (6,0)	0,9346 (5,0)	0,8399 (7,0)
DistalIPhalanxOutlineAgeGroup	0,7825 (1,0)	0,7650 (4,0)	0,7700 (2,0)	0,7550 (5,0)	0,6225 (7,0)	0,7700 (3,0)	0,7550 (6,0)
DistalIPhalanxOutlineCorrect	0,7517 (3,0)	0,7683 (1,0)	0,7550 (2,0)	0,7050 (7,0)	0,7483 (5,0)	0,7517 (4,0)	0,7250 (6,0)
DistalIPhalanxTW	0,7275 (2,0)	0,7325 (1,0)	0,7125 (4,0)	0,7100 (5,0)	0,6800 (7,0)	0,7225 (3,0)	0,7075 (6,0)
Earthquakes	0,6739 (7,0)	0,6957 (4,0)	0,7422 (3,0)	0,7516 (2,0)	0,8199 (1,0)	0,6863 (6,0)	0,6925 (5,0)
ECG200	0,8800 (3,5)	0,8800 (3,5)	0,8700 (5,0)	0,8900 (2,0)	0,8600 (6,0)	0,9000 (1,0)	0,7500 (7,0)
ECG5000	0,9249 (4,0)	0,9267 (1,0)	0,9260 (3,0)	0,9202 (5,0)	0,9151 (7,0)	0,9264 (2,0)	0,9189 (6,0)
ECGFiveDays	0,7967 (5,0)	0,9942 (1,0)	0,7642 (7,0)	0,8258 (4,0)	0,8432 (3,0)	0,7967 (6,0)	0,9199 (2,0)
ElectricDevices	0,5508 (3,0)	0,5417 (5,0)	0,5510 (2,0)	0,5255 (6,0)	0,5069 (7,0)	0,5595 (1,0)	0,5504 (4,0)
FaceAll	0,7136 (4,0)	0,8000 (1,0)	0,7651 (2,0)	0,7166 (3,0)	0,7018 (6,0)	0,7118 (5,0)	0,6941 (7,0)
FaceFour	0,7841 (3,0)	0,7614 (5,0)	0,7159 (7,0)	0,7273 (6,0)	0,8182 (2,0)	0,7841 (4,0)	0,8409 (1,0)
FacesUCR	0,7693 (6,0)	0,7878 (2,0)	0,7707 (5,0)	0,7751 (3,0)	0,7722 (4,0)	0,7654 (7,0)	0,8083 (1,0)
Fish	0,7829 (2,0)	0,7600 (6,0)	0,8000 (1,0)	0,7829 (3,0)	0,7657 (5,0)	0,7829 (4,0)	0,7143 (7,0)
FordA	0,6590 (5,0)	0,8689 (2,0)	0,6557 (6,0)	0,6823 (4,0)	0,6507 (7,0)	0,6840 (3,0)	0,8814 (1,0)
FordB	0,5578 (5,0)	0,8012 (2,0)	0,5473 (7,0)	0,5820 (4,0)	0,5558 (6,0)	0,5921 (3,0)	0,8215 (1,0)
GunPoint	0,9133 (4,0)	0,9667 (1,0)	0,9133 (5,0)	0,9067 (7,0)	0,9467 (2,0)	0,9133 (6,0)	0,9333 (3,0)
Ham	0,6000 (2,0)	0,5714 (4,0)	0,5619 (6,0)	0,5810 (3,0)	0,5714 (5,0)	0,6190 (1,0)	0,4952 (7,0)

HandOutlines	0,8010	(1,0)	0,7500	(7,0)	0,7850	(4,0)	0,7660	(6,0)	0,7890	(3,0)	0,8010	(2,0)	0,7670	(5,0)
Haptics	0,3701	(3,0)	0,4188	(1,0)	0,4026	(2,0)	0,3604	(6,0)	0,3636	(5,0)	0,3701	(4,0)	0,3474	(7,0)
Herring	0,5156	(3,0)	0,5000	(5,0)	0,6094	(2,0)	0,4219	(7,0)	0,4531	(6,0)	0,5156	(4,0)	0,6406	(1,0)
InlineSkate	0,3418	(1,0)	0,3091	(7,0)	0,3164	(4,0)	0,3109	(5,0)	0,3291	(3,0)	0,3418	(2,0)	0,3109	(6,0)
InsectWingbeatSound	0,5616	(2,0)	0,4753	(7,0)	0,5586	(4,0)	0,5535	(5,0)	0,5646	(1,0)	0,5591	(3,0)	0,5167	(6,0)
ItalyPowerDemand	0,9553	(4,0)	0,8571	(7,0)	0,9563	(3,0)	0,9572	(1,0)	0,9281	(5,0)	0,9572	(2,0)	0,9077	(6,0)
LargeKitchenAppliances	0,4933	(5,0)	0,7227	(2,0)	0,4907	(6,0)	0,5333	(4,0)	0,4453	(7,0)	0,5573	(3,0)	0,7733	(1,0)
Lightning2	0,7541	(5,5)	0,7541	(5,5)	0,7705	(3,5)	0,7705	(3,5)	0,8525	(1,0)	0,7541	(7,0)	0,8033	(2,0)
Lightning7	0,5753	(5,0)	0,5616	(7,0)	0,6027	(4,0)	0,6712	(2,0)	0,6849	(1,0)	0,5753	(6,0)	0,6712	(3,0)
MALLAT	0,9143	(2,0)	0,9203	(1,0)	0,9143	(3,0)	0,9134	(5,0)	0,7684	(7,0)	0,9143	(4,0)	0,8857	(6,0)
Meat	0,9333	(1,0)	0,9000	(5,0)	0,9333	(2,5)	0,9333	(2,5)	0,8833	(6,0)	0,9333	(4,0)	0,8667	(7,0)
MedicalImages	0,6842	(4,0)	0,6303	(7,0)	0,7013	(2,0)	0,6908	(3,0)	0,6658	(5,0)	0,7066	(1,0)	0,6526	(6,0)
MiddlePhalanxOutlineAgeGroup	0,7400	(2,0)	0,7425	(1,0)	0,7375	(3,0)	0,7175	(6,0)	0,6975	(7,0)	0,7300	(5,0)	0,7350	(4,0)
MiddlePhalanxOutlineCorrect	0,7533	(2,0)	0,7733	(1,0)	0,7533	(3,0)	0,7483	(4,0)	0,6217	(7,0)	0,7450	(5,0)	0,6750	(6,0)
MiddlePhalanxTW	0,5614	(5,0)	0,5714	(3,0)	0,5789	(1,0)	0,5764	(2,0)	0,5489	(7,0)	0,5689	(4,0)	0,5564	(6,0)
MoteStrain	0,8786	(2,0)	0,6765	(5,0)	0,8810	(1,0)	0,6685	(6,0)	0,7772	(4,0)	0,8746	(3,0)	0,6605	(7,0)
NonInvasiveFetalECG1	0,8290	(3,0)	0,7786	(5,0)	0,8214	(4,0)	0,8326	(2,0)	0,7735	(6,0)	0,8377	(1,0)	0,7471	(7,0)
NonInvasiveFetalECG2	0,8799	(1,0)	0,8514	(5,0)	0,8774	(4,0)	0,8794	(3,0)	0,8244	(6,0)	0,8799	(2,0)	0,8092	(7,0)
OliveOil	0,8667	(1,0)	0,8000	(6,0)	0,8333	(3,5)	0,8333	(3,5)	0,5333	(7,0)	0,8667	(2,0)	0,8333	(5,0)
OSULeaf	0,5207	(5,0)	0,6405	(1,0)	0,5372	(3,0)	0,5248	(4,0)	0,5165	(6,5)	0,5165	(6,5)	0,5826	(2,0)
PhalangesOutlinesCorrect	0,7611	(1,0)	0,7494	(4,0)	0,7564	(3,0)	0,7436	(5,0)	0,7133	(6,0)	0,7599	(2,0)	0,7040	(7,0)
Phoneme	0,1092	(4,0)	0,1588	(1,0)	0,1081	(5,0)	0,0649	(7,0)	0,0723	(6,0)	0,1139	(3,0)	0,1377	(2,0)
Plane	0,9619	(3,5)	0,9619	(3,5)	0,9714	(1,0)	0,9619	(5,0)	0,9714	(2,0)	0,9619	(6,5)	0,9619	(6,5)
ProximalPhalanxOutlineAgeGroup	0,7854	(2,0)	0,7659	(4,0)	0,7561	(7,0)	0,7756	(3,0)	0,7951	(1,0)	0,7659	(5,0)	0,7610	(6,0)
ProximalPhalanxOutlineCorrect	0,8076	(2,0)	0,8351	(1,0)	0,8076	(3,0)	0,7938	(4,0)	0,7526	(6,0)	0,7698	(5,0)	0,7354	(7,0)
ProximalPhalanxTW	0,7075	(2,5)	0,7075	(2,5)	0,6875	(5,0)	0,6775	(6,0)	0,6450	(7,0)	0,7050	(4,0)	0,7175	(1,0)
RefrigerationDevices	0,3947	(7,0)	0,4320	(3,0)	0,4213	(4,0)	0,4107	(5,0)	0,4347	(2,0)	0,4000	(6,0)	0,4400	(1,0)
Screen Type	0,3600	(4,0)	0,4080	(1,0)	0,3600	(5,0)	0,3947	(2,5)	0,3947	(2,5)	0,3600	(6,0)	0,3440	(7,0)
ShapeletSim	0,5389	(4,0)	0,6333	(2,0)	0,5278	(7,0)	0,5833	(3,0)	0,5389	(5,5)	0,5389	(5,5)	0,8667	(1,0)
ShapesAll	0,7517	(1,0)	0,7450	(5,0)	0,7517	(2,5)	0,7517	(2,5)	0,7450	(6,0)	0,7500	(4,0)	0,7200	(7,0)
SmallKitchenAppliances	0,3440	(6,0)	0,3520	(4,0)	0,3387	(7,0)	0,6373	(1,0)	0,5547	(2,0)	0,3467	(5,0)	0,4533	(3,0)
SonyAIBORobotSurface	0,6955	(5,0)	0,7121	(3,0)	0,6905	(6,0)	0,7205	(2,0)	0,6755	(7,0)	0,6988	(4,0)	0,7271	(1,0)
SonyAIBORobotSurfaceII	0,8594	(3,0)	0,8636	(1,0)	0,8363	(4,0)	0,8122	(5,0)	0,7545	(7,0)	0,8625	(2,0)	0,7880	(6,0)
StarLightCurves	0,8488	(5,0)	0,9496	(1,0)	0,8488	(6,0)	0,8953	(3,0)	0,8600	(4,0)	0,8488	(7,0)	0,9258	(2,0)
Strawberry	0,9380	(1,0)	0,9119	(6,0)	0,9217	(4,0)	0,9380	(2,0)	0,9168	(5,0)	0,9380	(3,0)	0,8646	(7,0)
SwedishLeaf	0,7888	(6,0)	0,8528	(1,0)	0,7920	(5,0)	0,8096	(4,0)	0,7456	(7,0)	0,8128	(3,0)	0,8384	(2,0)

Symbols	0,8995	(2,0)	0,8683	(6,0)	0,9075	(1,0)	0,8844	(4,0)	0,7990	(7,0)	0,8995	(3,0)	0,8704	(5,0)
SyntheticControl	0,8800	(5,0)	0,6233	(7,0)	0,9233	(4,0)	0,9900	(2,0)	0,9967	(1,0)	0,9867	(3,0)	0,6367	(6,0)
ToeSegmentation1	0,6798	(4,0)	0,7325	(2,0)	0,5833	(7,0)	0,6711	(6,0)	0,6886	(3,0)	0,6798	(5,0)	0,7982	(1,0)
ToeSegmentation2	0,8077	(4,0)	0,7846	(6,5)	0,7846	(6,5)	0,8154	(3,0)	0,8385	(1,0)	0,8077	(5,0)	0,8308	(2,0)
Trace	0,7600	(4,0)	0,8400	(2,0)	0,7200	(6,0)	0,7000	(7,0)	0,8200	(3,0)	0,7500	(5,0)	0,8800	(1,0)
TwoLeadECG	0,7471	(4,0)	0,8692	(2,0)	0,7480	(3,0)	0,6769	(7,0)	0,7436	(6,0)	0,7463	(5,0)	0,8788	(1,0)
TwoPatterns	0,9067	(4,0)	0,5072	(6,0)	0,9067	(5,0)	0,9477	(1,0)	0,9317	(3,0)	0,9360	(2,0)	0,5070	(7,0)
UWaveGestureLibraryAll	0,9481	(4,0)	0,7362	(7,0)	0,9436	(5,0)	0,9520	(1,0)	0,9486	(2,0)	0,9484	(3,0)	0,7962	(6,0)
uWaveGestureLibraryX	0,7393	(3,0)	0,4090	(7,0)	0,7370	(5,0)	0,7409	(1,0)	0,7379	(4,0)	0,7398	(2,0)	0,4961	(6,0)
uWaveGestureLibraryY	0,6616	(5,0)	0,3473	(7,0)	0,6697	(2,0)	0,6781	(1,0)	0,6619	(4,0)	0,6667	(3,0)	0,4065	(6,0)
uWaveGestureLibraryZ	0,6496	(4,0)	0,4422	(7,0)	0,6505	(3,0)	0,6533	(1,0)	0,6413	(5,0)	0,6508	(2,0)	0,4992	(6,0)
Wafer	0,9955	(2,0)	0,9946	(4,0)	0,9953	(3,0)	0,9958	(1,0)	0,9891	(7,0)	0,9942	(5,0)	0,9895	(6,0)
Wine	0,6111	(1,0)	0,5370	(5,0)	0,5556	(4,0)	0,5741	(3,0)	0,4815	(7,0)	0,6111	(2,0)	0,5370	(6,0)
WordsSynonyms	0,6176	(5,0)	0,4154	(7,0)	0,6191	(4,0)	0,6301	(1,0)	0,6270	(2,0)	0,6207	(3,0)	0,5345	(6,0)
Worms	0,3646	(5,0)	0,4696	(1,0)	0,3536	(7,0)	0,4033	(4,0)	0,4199	(2,0)	0,3646	(6,0)	0,4199	(3,0)
WormsTwoClass	0,5856	(4,0)	0,6630	(2,0)	0,5912	(3,0)	0,5801	(6,0)	0,5580	(7,0)	0,5856	(5,0)	0,6796	(1,0)
Yoga	0,8303	(3,0)	0,7920	(7,0)	0,8347	(1,0)	0,8267	(4,0)	0,8213	(5,0)	0,8307	(2,0)	0,8160	(6,0)
<i>Ranks médios</i>	3,42		3,69		3,88		3,76		4,68		3,75		4,82	

“Teste de Friedman para o primeiro experimento de ensembles”

Conjunto	DTW	Majoritário	Acurácia	Distância	P.P. (rank)	P.P. (peso)	Simple rank
50Words	0,7648 (2,0)	0,7349 (7,0)	0,7516 (4,0)	0,7780 (1,0)	0,7424 (6,0)	0,7549 (3,0)	0,7442 (5,0)
Adiac	0,6087 (7,0)	0,6980 (6,0)	0,7263 (1,0)	0,7008 (5,0)	0,7194 (2,0)	0,7138 (4,0)	0,7151 (3,0)
Beef	0,6667 (1,0)	0,5333 (4,0)	0,5333 (4,0)	0,5333 (4,0)	0,5333 (4,0)	0,5333 (4,0)	0,5133 (7,0)
CBF	0,9944 (1,0)	0,9597 (6,0)	0,9933 (2,0)	0,9211 (7,0)	0,9743 (3,0)	0,9700 (5,0)	0,9738 (4,0)
ChlorineConcentration	0,6500 (5,0)	0,6514 (4,0)	0,6526 (1,0)	0,6516 (3,0)	0,4759 (7,0)	0,6523 (2,0)	0,5679 (6,0)
CinCECGtorso	0,9304 (1,0)	0,9046 (4,0)	0,9246 (2,0)	0,8971 (7,0)	0,9243 (3,0)	0,9026 (6,0)	0,9046 (5,0)
Coffee	1,0000 (1,0)	0,8036 (6,0)	0,8214 (3,0)	0,7857 (7,0)	0,8071 (5,0)	0,8214 (4,0)	0,8571 (2,0)
CricketX	0,7718 (3,0)	0,7669 (5,0)	0,7769 (2,0)	0,7692 (4,0)	0,7526 (6,0)	0,7826 (1,0)	0,7497 (7,0)
CricketY	0,7615 (7,0)	0,7974 (4,0)	0,8077 (2,5)	0,8077 (2,5)	0,7813 (5,0)	0,8085 (1,0)	0,7805 (6,0)
CricketZ	0,7462 (7,0)	0,8236 (3,0)	0,8359 (1,0)	0,8282 (2,0)	0,8121 (5,0)	0,8203 (4,0)	0,7944 (6,0)
DiatomSizeReduction	0,9346 (7,0)	0,9428 (4,0)	0,9425 (5,0)	0,9575 (1,0)	0,9431 (3,0)	0,9412 (6,0)	0,9516 (2,0)
ECG200	0,8800 (4,0)	0,8720 (6,0)	0,8800 (5,0)	0,8500 (7,0)	0,9050 (2,0)	0,8940 (3,0)	0,9160 (1,0)
ECGFiveDays	0,7967 (7,0)	0,8728 (4,0)	0,9384 (2,0)	0,9512 (1,0)	0,8446 (5,0)	0,9023 (3,0)	0,8026 (6,0)

FaceAll	0,8083 (7,0)	0,8116 (6,0)	0,8178 (4,0)	0,8160 (5,0)	0,8209 (3,0)	0,8329 (1,0)	0,8249 (2,0)
FaceFour	0,8864 (1,0)	0,8375 (5,0)	0,8409 (4,0)	0,8523 (2,0)	0,8227 (6,0)	0,8455 (3,0)	0,7682 (7,0)
FacesUCR	0,9122 (1,0)	0,8946 (6,0)	0,9063 (3,0)	0,9088 (2,0)	0,8971 (5,0)	0,9060 (4,0)	0,8934 (7,0)
Fish	0,8457 (7,0)	0,8880 (3,0)	0,8743 (5,0)	0,8857 (4,0)	0,8914 (1,0)	0,8891 (2,0)	0,8709 (6,0)
GunPoint	0,9133 (6,0)	0,9420 (4,0)	0,9200 (5,0)	0,9667 (1,0)	0,9493 (3,0)	0,9540 (2,0)	0,9107 (7,0)
Haptics	0,4123 (7,0)	0,4289 (5,0)	0,4351 (4,0)	0,4286 (6,0)	0,4545 (2,0)	0,4555 (1,0)	0,4542 (3,0)
InlineSkate	0,3873 (4,0)	0,3795 (5,0)	0,3909 (2,5)	0,3909 (2,5)	0,3375 (7,0)	0,3964 (1,0)	0,3465 (6,0)
ItalyPowerDemand	0,9553 (7,0)	0,9573 (4,0)	0,9563 (5,0)	0,9582 (1,0)	0,9575 (3,0)	0,9577 (2,0)	0,9553 (6,0)
Lightning2	0,8689 (2,0)	0,8361 (5,0)	0,8689 (1,0)	0,8361 (6,0)	0,8459 (4,0)	0,8574 (3,0)	0,7639 (7,0)
Lightning7	0,7123 (7,0)	0,7493 (6,0)	0,7534 (5,0)	0,7808 (2,0)	0,7712 (4,0)	0,7726 (3,0)	0,8356 (1,0)
MALLAT	0,9143 (6,0)	0,9235 (4,0)	0,9177 (5,0)	0,9330 (1,0)	0,9267 (3,0)	0,9305 (2,0)	0,9136 (7,0)
MedicalImages	0,7474 (3,0)	0,7296 (6,0)	0,7368 (4,0)	0,7487 (1,0)	0,7224 (7,0)	0,7483 (2,0)	0,7301 (5,0)
MoteStrain	0,8658 (5,0)	0,8672 (4,0)	0,8786 (2,0)	0,8283 (7,0)	0,8722 (3,0)	0,8800 (1,0)	0,8486 (6,0)
NonInvasiveFetalECG1	0,8107 (6,0)	0,8496 (3,0)	0,8539 (1,0)	0,8499 (2,0)	0,0248 (7,0)	0,8479 (5,0)	0,8491 (4,0)
NonInvasiveFetalECG2	0,8748 (7,0)	0,9070 (4,0)	0,9120 (2,0)	0,9053 (5,0)	0,9025 (6,0)	0,9159 (1,0)	0,9088 (3,0)
OliveOil	0,8667 (2,0)	0,8333 (5,0)	0,8333 (5,0)	0,8333 (5,0)	0,8333 (5,0)	0,8333 (5,0)	0,8667 (1,0)
OSULeaf	0,6116 (7,0)	0,6149 (5,0)	0,6322 (2,0)	0,6364 (1,0)	0,6161 (3,0)	0,6153 (4,0)	0,6140 (6,0)
SonyAIBORobotSurface	0,6955 (5,0)	0,7126 (2,0)	0,7005 (3,0)	0,7288 (1,0)	0,6965 (4,0)	0,6860 (6,0)	0,5749 (7,0)
SonyAIBORobotSurfaceII	0,8594 (5,0)	0,8624 (3,0)	0,8625 (2,0)	0,8636 (1,0)	0,8399 (6,0)	0,8612 (4,0)	0,8085 (7,0)
StarLightCurves	0,8488 (7,0)	0,9208 (5,0)	0,9326 (1,0)	0,9271 (3,0)	0,9268 (4,0)	0,9292 (2,0)	0,9141 (6,0)
SwedishLeaf	0,8464 (7,0)	0,8914 (3,0)	0,8976 (2,0)	0,9024 (1,0)	0,8829 (6,0)	0,8901 (4,0)	0,8835 (5,0)
Symbols	0,9377 (5,0)	0,9397 (3,0)	0,9473 (2,0)	0,9528 (1,0)	0,9335 (6,0)	0,9390 (4,0)	0,9261 (7,0)
SyntheticControl	0,9833 (6,0)	0,9887 (3,0)	0,9933 (1,0)	0,8767 (7,0)	0,9867 (4,0)	0,9933 (2,0)	0,9867 (5,0)
Trace	0,9900 (1,0)	0,8860 (5,0)	0,9400 (3,0)	0,9800 (2,0)	0,8950 (4,0)	0,8830 (6,5)	0,8830 (6,5)
TwoLeadECG	0,8683 (6,0)	0,8764 (3,0)	0,8745 (4,0)	0,8894 (1,0)	0,8774 (2,0)	0,8719 (5,0)	0,7989 (7,0)
TwoPatterns	0,9972 (2,0)	0,9747 (6,0)	0,9980 (1,0)	0,9075 (7,0)	0,9940 (3,0)	0,9912 (4,0)	0,9900 (5,0)
uWaveGestureLibraryX	0,7733 (5,0)	0,7744 (4,0)	0,7753 (3,0)	0,7490 (7,0)	0,7550 (6,0)	0,7768 (2,0)	0,7839 (1,0)
uWaveGestureLibraryY	0,6991 (1,0)	0,6882 (5,0)	0,6926 (2,0)	0,6508 (6,0)	0,6417 (7,0)	0,6911 (3,0)	0,6906 (4,0)
uWaveGestureLibraryZ	0,6778 (5,0)	0,6795 (4,0)	0,6882 (1,0)	0,6602 (6,0)	0,5295 (7,0)	0,6822 (3,0)	0,6872 (2,0)
Wafer	0,9955 (7,0)	0,9967 (5,0)	0,9972 (1,0)	0,9968 (3,5)	0,9968 (3,5)	0,9971 (2,0)	0,9958 (6,0)
WordsSynonyms	0,7476 (1,0)	0,7118 (4,0)	0,7367 (3,0)	0,7414 (2,0)	0,6995 (6,0)	0,7118 (5,0)	0,6893 (7,0)
Yoga	0,8447 (6,0)	0,8574 (4,0)	0,8593 (3,0)	0,8633 (1,0)	0,8478 (5,0)	0,8614 (2,0)	0,8169 (7,0)
Ranks médios	4,60	4,49	2,80	3,43	4,48	3,17	5,03