



Pós-Graduação em Ciência da Computação

**“UMA ABORDAGEM ADAPTATIVA DE LEARNING  
VECTOR QUANTIZATION PARA CLASSIFICAÇÃO  
DE DADOS INTERVALARES”**

**Por**

***TELMO DE MENEZES E SILVA FILHO***

**Dissertação de Mestrado**



Universidade Federal de Pernambuco  
posgraduacao@cin.ufpe.br  
www.cin.ufpe.br/~posgraduacao

RECIFE, MARÇO/2013



UNIVERSIDADE FEDERAL DE PERNAMBUCO

CENTRO DE INFORMÁTICA

PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

TELMO DE MENEZES E SILVA FILHO

“UMA ABORDAGEM ADAPTATIVA DE LEARNING VECTOR  
QUANTIZATION PARA CLASSIFICAÇÃO DE DADOS INTERVALARES”

*ESTE TRABALHO FOI APRESENTADO À PÓS-GRADUAÇÃO EM  
CIÊNCIA DA COMPUTAÇÃO DO CENTRO DE INFORMÁTICA DA  
UNIVERSIDADE FEDERAL DE PERNAMBUCO COMO REQUISITO  
PARCIAL PARA OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIA DA  
COMPUTAÇÃO.*

ORIENTADORA: RENATA MARIA CARDOSO RODRIGUES DE SOUZA

RECIFE, MARÇO/2013

**Catálogo na fonte**  
**Bibliotecária Jane Souto Maior, CRB4-571**

**Silva Filho, Telmo de Menezes e**  
**Uma abordagem adaptativa de learning vector**  
**quantization para classificação de dados intervalares. /**  
**Telmo de Menezes e Silva Filho. - Recife: O Autor, 2013.**  
**xii, 61 folhas: il., fig., tab.**

**Orientadora: Renata Maria Cardoso Rodrigues de Souza.**  
**Dissertação (mestrado) - Universidade Federal de**  
**Pernambuco. Cln, Ciência da Computação, 2013.**

**Inclui bibliografia.**

**1. Inteligência computacional. 2. Redes neurais. 3. Dados**  
**simbólicos. I. Souza, Renata Maria Cardoso Rodrigues de**  
**(orientadora). II. Título.**

**006.3**

**CDD (23. ed.)**

**MEI2013 – 039**

Dissertação de Mestrado apresentada por TELMO DE MENEZES E SILVA FILHO à Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, sob o título “**Uma Abordagem Adaptativa de Learning Vector Quantization para Classificação de Dados Intervalares**” orientada pela Profa. RENATA MARIA CARDOSO RODRIGUES DE SOUZA e aprovada pela Banca Examinadora formada pelos professores:

---

Prof. Ricardo Bastos Cavalcante Prudêncio  
Centro de Informática / UFPE

---

Prof. Marco Antonio de Oliveira Domingues  
Departamento de Gestão de Tecnologia da Informação/IFPE

---

Profa. Renata Maria Cardoso Rodrigues de Souza  
Centro de Informática / UFPE

Visto e permitida a impressão.  
Recife, 27 de fevereiro de 2013

---

**Profa. Edna Natividade da Silva Barros**  
Coordenadora da Pós-Graduação em Ciência da Computação do  
Centro de Informática da Universidade Federal de Pernambuco.

*Dedico esta dissertação a meus pais, minha família, meus amigos, minha orientadora e a todos que me ajudaram em algum momento a concluir esta etapa da minha vida.*

# Agradecimentos

Louvo a Deus pelos meus pais,  
uma bênção verdadeira.  
Incondicional amor,  
zelo na dor passageira.

A minha família,  
parte de quem sou,  
é peça fundamental,  
se estou onde estou.

Sempre devo dizer  
obrigado às minhas irmãs.  
Agradeço com prazer.  
Vou ser sempre um grande fã.

Orientadora excelente,  
compreensiva me escutou,  
exigente e atenciosa e  
que tanto me ajudou.

Uns em Recife ou no Ceará.  
Em Penedo ou em qualquer lugar.  
Reside a amizade, nos  
castelos que ela formar.

A minha educação,  
se em parte devo a mim,  
antes devo a meus mestres,  
reis do saber sem fim.

Canto essas pessoas.  
Ofereço minha gratidão.  
Me ensinaram coisas boas,  
inscritas em meu coração,  
gravadas em minha lembrança,  
origens da minha formação.

*Nada melhor pode dar um pai a seu filho do que  
uma boa educação.*

—TEXTOS ISLÂMICOS

# Resumo

A Análise de Dados Simbólicos lida com tipos de dados complexos, capazes de modelar a variabilidade interna dos dados e dados imprecisos. Dados simbólicos intervalares surgem naturalmente de valores como variação de temperatura diária, pressão sanguínea, entre outros. Esta dissertação introduz um algoritmo de Learning Vector Quantization para dados simbólicos intervalares, que usa uma distância Euclidiana intervalar ponderada e generalizada para medir a distância entre instâncias de dados e protótipos.

A distância proposta tem quatro casos especiais. O primeiro caso é a distância Euclidiana intervalar e tende a modelar classes e clusters com formas esféricas. O segundo caso é uma distância intervalar baseada em protótipos que modela subregiões não-esféricas e de tamanhos similares dentro das classes. O terceiro caso permite à distância lidar com subregiões não-esféricas e de tamanhos variados dentro das classes. O último caso permite à distância modelar classes desbalanceadas, compostas de subregiões de várias formas e tamanhos. Experimentos são feitos para avaliar os desempenhos do Learning Vector Quantization intervalar proposto, usando todos os quatro casos da distância proposta. Três conjuntos de dados intervalares sintéticos e um conjunto de dados intervalares reais são usados nesses experimentos e seus resultados mostram a utilidade de uma distância localmente ponderada.

**Palavras-chave:** Análise de Dados Simbólicos, Learning Vector Quantization, Distância Ponderada

# Abstract

Symbolic Data Analysis deals with complex data types, capable of modeling internal data variability and imprecise data. Symbolic interval data comes naturally from the description of ranges of values, such as daily temperature variation, blood pressure, etc. This dissertation introduces a Learning Vector Quantization algorithm for symbolic interval data, that uses a generalized weighted interval Euclidean distance to measure the distance between data instances and prototypes.

The proposed distance has four special cases. The first case is the interval Euclidean distance and tends to model spherical classes and clusters. The second case is a prototype-based interval distance that models non-spherical and similarly sized subregions within classes. The third case allows the distance to deal with non-spherical subregions of varied sizes within classes. The last case allows the distance to model imbalanced classes composed of subregions with varied shapes and sizes. Experiments are made to assess the performances of the proposed interval Learning Vector Quantization, using all four cases of the proposed distance. Three synthetic interval datasets and one real interval dataset are used in these experiments and their results show the usefulness of a locally weighted distance.

**Keywords:** Symbolic Data Analysis, Learning Vector Quantization, Weighted Distance

# Sumário

<b>Lista de Figuras</b>	<b>x</b>
<b>Lista de Tabelas</b>	<b>xi</b>
<b>Lista de Acrônimos</b>	<b>xii</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Motivação . . . . .	2
1.2 Trabalhos Relacionados . . . . .	3
1.2.1 Extensões de Métodos para Dados Simbólicos . . . . .	3
1.2.2 A Família LVQ . . . . .	4
1.3 Contribuição . . . . .	5
1.4 Estrutura . . . . .	7
<b>2 Dados Simbólicos</b>	<b>8</b>
2.1 Tipos de Dados . . . . .	9
2.1.1 Variáveis Multivaloradas . . . . .	12
2.1.2 Variáveis Intervalares . . . . .	12
2.1.3 Variáveis Modais . . . . .	13
<b>3 Aprendizagem Baseada em Protótipos</b>	<b>14</b>
3.1 k-Vizinhos mais Próximos . . . . .	15
3.2 Versões otimizadas do k-NN . . . . .	15
3.2.1 k-Vizinhos Mais Próximos Editados . . . . .	15
3.2.2 k-Vizinhos Mais Próximos Condensados . . . . .	16
3.2.3 Links de Tomek . . . . .	18
3.2.4 Seleção Unilateral . . . . .	18
3.3 k-NN Simbólico . . . . .	19
3.4 Classificador baseado em k-Médias . . . . .	20
3.4.1 c-Médias Difuso Intervalar . . . . .	21
<b>4 O LVQ</b>	<b>22</b>
4.1 Decisão Ótima . . . . .	22
4.2 O LVQ1 . . . . .	23
4.3 O LVQ1 Otimizado . . . . .	24

---

4.4	Outras variações . . . . .	25
4.4.1	O LVQ2 . . . . .	25
4.4.2	O LVQ3 . . . . .	26
4.4.3	O LVQ1-Batch . . . . .	27
4.5	Versões Adaptativas . . . . .	27
4.5.1	OLVQ-AEDC . . . . .	28
	O algoritmo . . . . .	29
4.5.2	OLVQ-AEDP . . . . .	30
	O algoritmo . . . . .	32
<b>5</b>	<b>OLVQ Intervalar</b>	<b>34</b>
5.1	Casos da Distância Ponderada Generalizada para Dados Intervalares . . . . .	35
5.2	Distância Quadrática Euclidiana Ponderada Generalizada para Dados Intervalares . . . . .	36
5.3	O Algoritmo . . . . .	37
5.3.1	O bloco de passos de treinamento online . . . . .	37
5.3.2	O bloco batch de ajuste de pesos . . . . .	38
5.3.3	O bloco de passos de validação . . . . .	39
5.3.4	Passos do algoritmo . . . . .	39
<b>6</b>	<b>Análise de Desempenho</b>	<b>42</b>
6.1	Resultados dos Conjuntos de Dados Sintéticos . . . . .	42
6.1.1	Primeiro Conjunto de Dados Intervalares Sintéticos . . . . .	43
6.1.2	Segundo Conjunto de Dados Intervalares Sintéticos . . . . .	46
6.1.3	Terceiro Conjunto de Dados Intervalares Sintéticos . . . . .	49
6.2	Resultados do Conjunto de Dados de Clima . . . . .	52
<b>7</b>	<b>Conclusões</b>	<b>57</b>
7.1	Trabalhos futuros . . . . .	58
	<b>Referências Bibliográficas</b>	<b>59</b>

---

# Lista de Figuras

3.1	Exemplo de k-NN, classe do novo elemento será definida como a classe mais comum entre os $k$ vizinhos mais próximos. . . . .	16
3.2	Exemplo de ENN. . . . .	17
3.3	Resultado do ENN. . . . .	17
3.4	Conjunto de dados original com ruído e desbalanceamento entre as classes. . . . .	19
3.5	Conjunto após a aplicação do CNN, ainda com ruído entre as classes. . . . .	19
3.6	Resultado do OSS, após a eliminação dos links de Tomek. . . . .	19
4.1	Exemplo de LVQ2. . . . .	26
4.2	Exemplo de classe formada por intra-regiões. . . . .	31
6.1	Primeiro conjunto de dados sintéticos, considerando o intervalo [1,10]. . . . .	44
6.2	Primeiro conjunto de dados sintéticos, considerando o intervalo [1,20]. . . . .	44
6.3	Primeiro conjunto de dados sintéticos, considerando o intervalo [1,30]. . . . .	44
6.4	Primeiro conjunto de dados sintéticos, considerando o intervalo [1,40]. . . . .	44
6.5	Segundo conjunto de dados sintéticos, considerando o intervalo [1,10]. . . . .	47
6.6	Segundo conjunto de dados sintéticos, considerando o intervalo [1,20]. . . . .	47
6.7	Segundo conjunto de dados sintéticos, considerando o intervalo [1,30]. . . . .	47
6.8	Segundo conjunto de dados sintéticos, considerando o intervalo [1,40]. . . . .	47
6.9	Terceiro conjunto de dados sintéticos, considerando o intervalo [1,10]. . . . .	50
6.10	Terceiro conjunto de dados sintéticos, considerando o intervalo [1,20]. . . . .	50
6.11	Terceiro conjunto de dados sintéticos, considerando o intervalo [1,30]. . . . .	50
6.12	Terceiro conjunto de dados sintéticos, considerando o intervalo [1,40]. . . . .	50

# Lista de Tabelas

2.1	Identificação das variáveis. . . . .	9
2.2	Dados Clássicos. . . . .	10
2.3	Dados Simbólicos. . . . .	11
2.4	Dados Simbólicos. . . . .	11
6.1	Parâmetros das distribuições Gaussianas . . . . .	43
6.2	Média e desvio padrão da taxa de erros de classificação, em %, e número de passos de treinamento para o primeiro conjunto de dados sintéticos . . . . .	45
6.3	Estatísticas dos testes de hipóteses para a taxa de erros de classificação do primeiro conjunto de dados sintéticos . . . . .	46
6.4	Estatísticas dos testes de hipóteses para o número de passos do primeiro conjunto de dados sintéticos . . . . .	46
6.5	Parâmetros das distribuições Gaussianas . . . . .	46
6.6	Média e desvio padrão da taxa de erros de classificação, em %, e número de passos de treinamento para o segundo conjunto de dados sintéticos . . . . .	48
6.7	Estatísticas dos testes de hipóteses para a taxa de erros de classificação do segundo conjunto de dados sintéticos . . . . .	49
6.8	Estatísticas dos testes de hipóteses para o número de passos do segundo conjunto de dados sintéticos . . . . .	49
6.9	Parâmetros das distribuições Gaussianas . . . . .	49
6.10	Média e desvio padrão da taxa de erros de classificação, em %, e número de passos de treinamento para o terceiro conjunto de dados sintéticos . . . . .	51
6.11	Estatísticas dos testes de hipóteses para a taxa de erros de classificação do terceiro conjunto de dados sintéticos . . . . .	52
6.12	Estatísticas dos testes de hipóteses para o número de passos do terceiro conjunto de dados sintéticos . . . . .	52
6.13	Média e desvio padrão da taxa de erros de classificação, em %, e número de passos de treinamento para o conjunto de dados de clima . . . . .	53
6.14	Estatísticas dos testes t de Student e Kruskal-Wallis para as médias de taxa de erros e número de passos, respectivamente. . . . .	54
6.15	Amostra do conjunto de dados climáticos. . . . .	55
6.16	Amostra do conjunto de dados climáticos. . . . .	56

# Lista de Acrônimos

<b>LVQ</b>	Learning Vector Quantization
<b>OLVQ</b>	Optimized Learning Vector Quantization
<b>ADS</b>	Análise de Dados Simbólicos
<b>GVM</b>	Grafo de Vizinhança Mútua
<b>GRLVQ</b>	Generalized Relevance Learning Vector Quantization
<b>ADS</b>	Análise de Dados Simbólicos
<b>SOM</b>	Self-organizing Maps
<b>OLVQ-WD</b>	Optimized Learning Vector Quantization with a Weighted Distance
<b>OLVQ-AEDC</b>	Optimized Learning Vector Quantization with an Adaptive Euclidean Distance by Class
<b>OLVQ-AEDP</b>	Optimized Learning Vector Quantization with an Adaptive Euclidean Distance by Prototype
<b>IOLVQ</b>	Interval Optimized Learning Vector Quantization
<b>AS</b>	Aprendizagem Supervisionada
<b>MLP</b>	Multilayer Perceptron
<b>iMLP</b>	interval Multilayer Perceptron
<b>SVM</b>	Support Vector Machines
<b>k-NN</b>	k-Vizinhos mais Próximos, do inglês k-Nearest Neighbors
<b>ENN</b>	Edited Nearest Neighbors
<b>CNN</b>	Condensed Nearest Neighbors
<b>OLD-SKNN</b>	Optimal Local Distance-Weighted Symbolic k-NN
<b>OSS</b>	One-Sided Selection

# 1

## Introdução

*Vocês não sabem de nada!*

*Vocês não lêem!*

—AFONSO CARLOS RODRIGUES TIMÓTEO

Instâncias de dados clássicos são geralmente definidas como vetores de variáveis quantitativas e/ou qualitativas. Devido a isso, a análise de dados clássicos não modela naturalmente variabilidade e incerteza para representar dados de tipos complexos. A Análise de Dados Simbólicos (ADS) introduz uma variedade de tipos de dados que representam melhor variabilidade de dados, como intervalos, histogramas, listas de valores e outros (Diday e Noirhomme-Fraiture, 2008).

O interesse em dados intervalares tem crescido com os recentes avanços em tecnologias de bancos de dados e inteligência computacional. Esse tipo de dados é principalmente estudado pela ADS, que é um domínio da área de mineração e gerenciamento de dados, relacionado à análise multivariada, reconhecimento de padrões e inteligência artificial. Este trabalho foca nos dados simbólicos intervalares. Esse tipo de dados surge naturalmente da descrição de intervalos de valores, como variação da temperatura diária, preços diários de ações da bolsa de valores, etc.

O restante desse capítulo descreve o foco dessa dissertação e começa apresentando sua motivação na Seção 1.1. A Seção 1.2 explica alguns trabalhos relacionados a essa dissertação. A Seção 1.3 apresenta trabalhos relacionados e as principais contribuições e,

por último, Seção 1.4 descreve como a dissertação está organizada.

## 1.1 Motivação

Dados simbólicos são capazes de sumarizar grandes bases de dados clássicos em novos conjuntos de dados simbólicos de tamanho menor, tornando mais ágil sua análise. Os dados presentes numa base simbólica retêm as informações contidas numa base de dados clássica apresentado-as em conjunto menor e, na grande maioria das vezes, com perda mínima de representatividade.

Para a área da Inteligência Computacional a utilização desse tipo de dados é bastante interessante, devido a sua capacidade de modelar incerteza e variabilidade, sendo necessário o desenvolvimento de novos algoritmos que sejam capazes de lidar com dados simbólicos ou a adaptação de métodos consolidados de dados clássicos. Vários métodos desenvolvidos originalmente para dados clássicos já foram adaptados para dados simbólicos, como multilayer perceptrons, regressões, k-vizinhos mais próximos, entre outros.

Um algoritmo que demonstra potencial para ser estendido para lidar com dados intervalares, devido a sua simplicidade e eficiência, é o Learning Vector Quantization (LVQ). LVQ é um algoritmo baseado em protótipos proposto por Kohonen (Kohonen, 2001). O algoritmo inicia designando aleatoriamente um subconjunto de protótipos para cada classe do conjunto de dados. Então os protótipos são iterativamente atualizados, tal que a regra do vizinho mais próximo minimize a probabilidade esperada de classificações incorretas. Ao final das iterações, os protótipos atualizados devem estar próximos aos padrões de suas classes. A extensão do LVQ para dados simbólicos se trata de substituir a distância utilizada por uma distância para dados simbólicos. Tipicamente se utiliza a distância Euclidiana para os métodos LVQ clássicos, portanto pode-se utilizar a distância Euclidiana intervalar (de Carvalho *et al.*, 2006) para um LVQ intervalar.

O problema da distância Euclidiana é que ela não modela classes e clusters de formas variadas, ao contrário das distâncias ponderadas. Distâncias ponderadas ou adaptativas são medidas utilizadas com o intuito de modelar características presentes nos dados, como clusters ou classes de diversas formas e tamanhos. Para atingir esse objetivo, essas distâncias utilizam pesos que podem ser atribuídos, por exemplo, às variáveis, aos protótipos ou às classes. Experimentos com dados clássicos tem demonstrado bons resultados para o uso de distâncias ponderadas ou adaptativas (Silva Filho e Souza, 2011; Paredes e Vidal, 2004; Borgelt *et al.*, 2005; Schneider *et al.*, 2009; Karayiannis e Randolph-Gips, 2003). Além disso, distâncias adaptativas também já se mostraram úteis

para dados simbólicos (de Carvalho, 2007).

Assim, torna-se interessante o desenvolvimento de um algoritmo LVQ para dados simbólicos, que utilize uma distância ponderada para modelar características inerentes aos dados.

## 1.2 Trabalhos Relacionados

Existem vários trabalhos relacionados a essa dissertação, seja porque adaptaram para dados simbólicos métodos clássicos já existentes na literatura ou porque sugeriram mudanças ao algoritmo LVQ padrão, como a incorporação de métricas mais flexíveis ou busca dinâmica do número ótimo de protótipos. Alguns desses trabalhos estão listados a seguir.

### 1.2.1 Extensões de Métodos para Dados Simbólicos

Diversos algoritmos de classificação clássicos foram estendidos para lidar com dados simbólicos, como redes neurais, classificadores orientados a regiões, classificadores baseados em abordagens de regressão e k-vizinhos mais próximos.

Na área de redes neurais, algumas variações foram propostas que utilizam entradas, saídas ou pesos intervalares. Seguindo essa ideia, Ishibuchi *et al.* (1993) propuseram uma rede cujos pesos, biases e neurônios da camada de saída são intervalares, mas a entrada é clássica. Já na rede proposta por Beheshti *et al.* (1998), tanto os valores de entrada quanto os pesos, biases e saídas são intervalares. Além disso, Beheshti *et al.* (1998) mostram como obter pesos e biases ótimos para um dado conjunto de treinamento. Ainda para lidar com entradas e saídas intervalares, Rossi e Conan-guez (2002) e Roque *et al.* (2007) introduziram abordagens de MLP que usam pesos e bias clássicos e tem como vantagem a simplicidade de sua arquitetura, em comparação com redes neurais que utilizam pesos e bias intervalares.

Ichino *et al.* (1996) introduziram uma abordagem orientada a regiões para dados multivalorados. Nessa abordagem, as classes de exemplos são descritas por uma região (ou conjunto de regiões) obtida através do uso de uma aproximação de um Grafo de Vizinhança Mútua (GVM) e de um operador de junção simbólico. Outra abordagem orientada a regiões foi apresentada por D'Oliveira *et al.* (2004), para classificação de imagens de Radar de Abertura Sintética, em que cada região é definida pela envoltória convexa dos objetos que pertencem a uma classe.

Na área de regressão, Silva e Brito (2006) propuseram três abordagens para a análise multivariada de dados intervalares, focando na análise de discriminante linear e Souza *et al.* (2011) introduziram quatro classificadores de padrões diferentes baseados na metodologia de regressão logística que diferem em relação à maneira como representam os intervalos.

Para lidar com dados intervalares e modais Appice *et al.* (2006) introduziram uma abordagem de lazy learning, chamada Optimal Local Distance-Weighted Symbolic k-NN (OLD-SKNN), que estende um k-vizinhos mais próximos ponderado por distância clássico. O OLD-SKNN escolhe dinamicamente o número k de vizinhos mais próximos e dá como resposta as probabilidades de cada classe para o elemento de entrada.

### 1.2.2 A Família LVQ

A família do algoritmo LVQ consiste de diversas modificações ao procedimento básico do LVQ, com o objetivo de obter uma convergência mais rápida ou mais robusta e a incorporação de métricas mais flexíveis.

Kietzmann *et al.* (2008) apresentaram uma nova versão de Generalized Relevance Learning Vector Quantization (GRLVQ), com protótipos adicionados incrementalmente, usada para o caso não-trivial de reconhecimento de objetos de alta dimensionalidade num cenário de visão computacional. Também com uma forma de buscar dinamicamente o número ótimo de protótipos, Paredes e Vidal (2004) propuseram um algoritmo LVQ com redução de protótipos que emprega pesos para os protótipos. O uso de pesos também está presente no método de Borgelt *et al.* (2005), que é um LVQ difuso com uma distância adaptativa, permitindo que o método encontre clusters (classes) de diferentes formas e tamanhos. Ainda nessa linha de trabalho, Schneider *et al.* (2009) propuseram métodos adaptativos, no esforço de melhorar a performance com clusters anisotrópicos e Karayiannis e Randolph-Gips (2003) propuseram algoritmos soft de agrupamento e LVQ que utilizam múltiplas normas ponderadas não-Euclidianas para medir a distância entre vetores de dados e seus protótipos. Distâncias adaptativas também aparecem no trabalho de Silva Filho e Souza (2011), onde foram introduzidos dois algoritmos baseados no Optimized Learning Vector Quantization (OLVQ) que usam distâncias adaptativas, sendo que um dos algoritmos utiliza uma distância adaptativa baseada em classes, que pretende modelar classes de diversos formatos e o outro algoritmo usa uma distância Euclidiana baseada em protótipos para modelar classes compostas de sub-regiões de diferentes formas.

## 1.3 Contribuição

Como resultado do trabalho apresentado nessa dissertação, as seguintes contribuições podem ser destacadas:

**Análise de dados simbólicos.** Descrição e explicação de alguns dos tipos de dados simbólicos, incluindo dados intervalares, que são o foco desse trabalho. Para essa explicação, foi usado um conjunto de dados clássicos de exemplo. Esse conjunto foi agregado para a formação de um conjunto de dados simbólicos dos tipos multivalorados, intervalos e modais.

Valores de variáveis multivaloradas podem ser listas de valores categóricos e quantitativos, como  $Y_1 = azul, verde, branco$ . Valores de variáveis intervalares, como o próprio nome diz, são intervalos numéricos, como  $Y_2 = [1, 3]$ . Valores de variáveis modais são como os de variáveis multivaloradas, mas cada categoria possui um peso, por exemplo  $Y_3 = \{(azul, 1/3), (verde, 1/3), (branco, 1/3)\}$ .

A explicação dos dados simbólicos serve para demonstrar porque esses dados são importantes para a Inteligência Computacional e porque é interessante desenvolver um método LVQ para eles.

**Uma análise sobre algoritmos da família LVQ.** Apresenta uma análise breve de alguns algoritmos que compõem a família LVQ, além de explicar dois métodos que utilizam distâncias adaptativas para dados clássicos. Os métodos descritos são os clássicos LVQ1, LVQ1, LVQ3, OLVQ1 e LVQ1-Batch propostos por Kohonen (2001) e dois métodos adaptativos propostos por Silva Filho e Souza (2011), que mostraram bons resultados em seus experimentos e serviram de base para o algoritmo proposto nessa dissertação.

**O algoritmo proposto.** Apresenta o algoritmo LVQ intervalar e a distância intervalar ponderada e generalizada proposta. Essa distância é ponderada para modelar características inerentes aos dados. Para isso, essa distância tem três vetores de pesos:

1. cada protótipo tem um vetor de pesos  $\lambda$ , que modela a influência das variáveis na dispersão dos dados representados pelo protótipo (cada variável tem um peso correspondente no vetor de pesos de cada protótipo);
2. cada classe tem um vetor de pesos  $\gamma$  que representa a importância de cada protótipo para aquela classe (cada protótipo tem um peso correspondente no vetor de pesos da sua classe);
3. o conjunto de dados tem um vetor de pesos  $\beta$  que representa a importância de cada

classe no conjunto de dados (cada classe tem um peso correspondente no vetor de pesos do conjunto de dados).

Considerando esses pesos, a distância apresentada tem quatro casos:

1. no primeiro caso, todos os pesos permanecem fixos e a distância modela classes e subregiões de classes de forma esférica;
2. no segundo caso, os pesos  $\lambda$  variam, permitindo à distância modelar subregiões de classes (representadas pelos protótipos) de formas diferentes e tamanhos iguais;
3. no terceiro caso, os pesos  $\lambda$  e  $\gamma$  variam e a distância torna-se capaz de modelar subregiões de classes com formas e tamanhos variados;
4. no quarto caso, todos os pesos variam. Com isso, a distância pode modelar classes desbalanceadas e compostas de subregiões de variadas formas e tamanhos.

**Estudos empíricos para validar o método proposto.** Essa dissertação também apresenta um estudo de desempenho para o algoritmo proposto, considerando os quatro casos da distância ponderada e generalizada. A avaliação de desempenho é feita com um conjunto de dados intervalares reais de climas, obtido a partir de informações meteorológicas oficiais de várias cidades do mundo, e três conjuntos de dados intervalares sintéticos que demonstram as características modeladas pelos três casos mais gerais da distância: o primeiro conjunto apresenta classes compostas de subregiões de várias formas e tamanhos iguais (caso 2), o segundo conjunto apresenta classes compostas por subregiões de formas e tamanhos variados (caso 3) e o terceiro conjunto apresenta classes desbalanceadas compostas por subregiões de formas e tamanhos variados (caso 4).

**Produção científica.** Além das contribuições listadas acima, foram produzidos alguns artigos no processo de construção dessa dissertação:

- Silva Filho, T. M. e Souza, R. M. C. R. (2011). Pattern Classifiers with Adaptive Distances. International Joint Conference on Neural Networks (IJCNN), pages 1508-1514.
- Silva Filho, T. M. e Souza, R. M. C. R. (2012). A Weighted Learning Vector Quantization Approach for Interval Data. 19th international conference on Neural Information Processing (ICONIP), pages 504-511.

## 1.4 Estrutura

Essa dissertação está organizada da seguinte maneira: O Capítulo 2 traz uma explicação sobre dados simbólicos, incluindo dados multivalorados, intervalares e modais. O Capítulo 3 lista algoritmos supervisionados baseados em protótipos clássicos e simbólicos, relacionados ao LVQ. O Capítulo 4 analisa algoritmos da família LVQ, incluindo dois métodos adaptativos desenvolvidos ao longo dos estudos que levaram a essa dissertação. O Capítulo 5 apresenta a distância ponderada e generalizada intervalar proposta nesse trabalho e o LVQ intervalar que utiliza a distância apresentada. O Capítulo 6 traz o estudo dos desempenhos do algoritmo proposto, comparando os quatro casos da distância ponderada e generalizada intervalar, utilizando conjuntos de dados intervalares sintéticos e reais. Por fim, o Capítulo 7 traz as considerações finais.

# 2

## Dados Simbólicos

*Não se tira nada de nada, o novo vem do antigo,  
mas nem por isso é menos novo.*

—BERTOLT BRECHT

Padrões de dados clássicos são usualmente definidos como vetores de variáveis quantitativas e/ou qualitativas. Devido a isso, a análise de dados clássicos não representa naturalmente variabilidade ou incerteza na representação de dados complexos. A Análise de Dados Simbólicos (ADS) introduz uma variedade de tipos de dados que representam melhor a variabilidade dos dados, e.g. histogramas, listas de valores, intervalos, entre outros (Diday e Noirhomme-Fraiture, 2008).

Suponha que seja necessário descrever peixes de uma determinada espécie. Pode-se dizer: “O comprimento é de 30 a 60 cm, mas não passa de 40 cm nas fêmeas”. Não é possível colocar esse tipo de informação em tabelas de dados clássicos, pois elas são constituídas de células com simples valores de pontos. Também não é fácil representar regras em tabelas clássicas. Assim, esses peixes são descritos corretamente da seguinte forma:

[Comprimento = [30, 60]], [Sexo = {macho, fêmea}]  
e [Se {Sexo = fêmea} Então {Comprimento <= 40}]

o objetivo da ADS é obter ferramentas que possam lidar com esses tipos complexos de dados.

De acordo com Diday e Billard (2006), de uma forma geral, um “ponto” de dados simbólicos assume valores como um hipercubo ou um produto Cartesiano de distribuições no espaço p-dimensional ou uma mistura de ambos, em contraste com um “ponto” de dados clássicos que assume um simples valor pontual no espaço p-dimensional. Ou seja,

variáveis simbólicas podem ter valores como listas, intervalos, etc.

Dados simbólicos podem surgir naturalmente de certas situações como o intervalo da variação da temperatura diária de uma certa cidade ou uma lista das categorias de uma certa população e suas respectivas freqüências. Eles também podem ser o resultado da agregação de uma base de dados através de categorias que constituam um nível mais alto de interesse do pesquisador. Por exemplo: a agregação de uma base de dados que se tratava inicialmente dos moradores de um país pode gerar uma nova base, cujas instâncias são as cidades do país. Essa agregação também pode ser efetuada porque o conjunto original era tão grande que tornava difícil sua análise.

## 2.1 Tipos de Dados

Suponha um conjunto de dados composto de registros dos moradores de um país, resultado de um censo demográfico. Suponha que, para cada indivíduo, tem-se o registro de sua região (N, NE, CO, SE ou S), sua cidade (Recife, Manaus, etc), naturalidade (brasileiro ou não), sexo (M ou F), profissão, renda anual e idade. Esse típico conjunto de dados pode ser representado pela Tabela 2.2. A Tabela 2.1 contém as descrições das variáveis aleatórias  $Y_1, Y_2, \dots$  usadas na Tabela 2.2.

**Tabela 2.1** Identificação das variáveis.

$Y_i$	Descrição: valores possíveis
$Y_1$	Região: Norte (N), Nordeste (NE), Centro-Oeste (CO), Sudeste (SE), Sul (S)
$Y_2$	Cidade
$Y_3$	Brasileiro: sim (S), não (N)
$Y_4$	Sexo: masculino (M), feminino (F)
$Y_5$	Profissão
$Y_6$	Renda anual (em milhares de reais): $\geq 0$
$Y_7$	Idade (em anos): $\geq 0$

As entradas da Tabela 2.2 são valores de dados clássicos, com cada linha correspondendo a valores observados para as variáveis  $Y = (Y_1, \dots, Y_p)$  para um indivíduo. Nesse conjunto existem apenas  $n = 40$  observações e  $p = 7$  variáveis, sendo possível analisar os dados com técnicas clássicas. Em situações mais extremas (e.g  $n = 200$  milhões,  $p = 150$ ) a análise clássica pode ser problemática. Assim, pode ser interessante analisar esses dados considerando certas categorias. Por exemplo, habitantes da região Sul, ou médicos da região Sudeste. Como cada uma dessas categorias consiste de vários

**Tabela 2.2** Dados Clássicos.

i	$Y_1$	$Y_2$	$Y_3$	$Y_4$	$Y_5$	$Y_6$	$Y_7$
1	N	Belém	S	M	Engenheiro	60	43
2	S	Porto Alegre	N	F	Advogada	84	36
3	S	Curitiba	S	M	Engenheiro	36	24
4	CO	Cuiabá	S	M	Agricultor	120	50
5	NE	Recife	S	F	Médica	140	47
6	N	Manaus	N	F	Médica	96	60
7	NE	Recife	S	M	Porteiro	8.4	25
8	NE	Fortaleza	S	F	Médica	108	40
9	N	Palmas	S	M	Estudante	3.6	18
10	SE	São Paulo	N	M	Advogado	240	55
11	SE	Vitória	S	F	Médica	120	38
12	SE	São Paulo	S	M	Médico	144	66
13	S	Curitiba	N	F	Engenheira	96	34
14	NE	Fortaleza	S	F	Engenheira	48	27
15	N	Belém	S	M	Médico	72	32
16	N	Belém	S	M	Porteiro	12	67
17	S	Porto Alegre	S	F	Engenheira	36	42
18	SE	São Paulo	S	F	Estudante	0	16
19	NE	Recife	N	M	Médico	180	58
20	S	Porto Alegre	N	M	Engenheiro	84	26
21	S	Curitiba	S	F	Porteira	24	41
22	NE	Recife	S	F	Médica	120	31
23	N	Manaus	S	M	Porteiro	8.4	25
24	SE	Vitória	S	M	Estudante	6	19
25	NE	Fortaleza	S	F	Estudante	0	20
26	N	Belém	S	F	Engenheira	96	42
27	SE	São Paulo	N	F	Engenheira	120	30
28	S	Porto Alegre	S	M	Médico	180	37
29	NE	Recife	S	F	Porteira	9.6	29
30	N	Belém	N	M	Médico	240	48
31	SE	Vitória	S	F	Engenheira	36	34
32	NE	Recife	S	M	Engenheiro	48	26
33	N	Manaus	S	F	Médica	108	51
34	S	Curitiba	S	M	Porteiro	24	28
35	SE	São Paulo	N	F	Advogada	120	64
36	NE	Recife	S	M	Advogado	120	41
37	NE	Fortaleza	S	F	Estudante	0	23
38	N	Belém	S	F	Advogada	84	33
39	S	Curitiba	S	M	Advogado	108	37
40	SE	Vitória	N	M	Advogado	72	29

indivíduos, o valor observado não é mais um simples ponto. Por exemplo, a idade dos habitantes da região Sul (segundo a Tabela 2.2) é a lista  $\{24, 26, 28, 34, 36, 37, 37, 41, 42\}$ . Esses valores também podem ser representados pelo intervalo  $[24, 42]$  ou pelo histograma  $\{[20, 25), 1/9; [25, 30), 2/9; [30, 35), 1/9; [35, 40), 3/9; [40, 45], 2/9\}$ . Assim, a variável idade, para a categoria “habitantes da região Sul”, assumiu valores como lista, intervalo e histograma e cada um desses valores representa um tipo de dado simbólico. Se tomarmos a variável profissão, teremos 11 categorias ou conceitos simbólicos. Esses conceitos formam a Tabela 2.3 e a Tabela 2.4.

**Tabela 2.3** Dados Simbólicos.

$w_u$	Profissão	$Y_2$
$w_1$	Médico	{Recife,Manaus,Fortaleza,Vitória,São Paulo,Belém,Porto Alegre}
$w_2$	Engenheiro	{Belém,Curitiba,Fortaleza,Porto Alegre,São Paulo,Vitória,Recife}
$w_3$	Advogado	{Porto Alegre,São Paulo,Recife,Belém,Curitiba,Vitória}
$w_4$	Porteiro	{Recife,Belém,Curitiba,Manaus}
$w_5$	Estudante	{Palmas,São Paulo,Vitória,Fortaleza}
$w_6$	Agricultor	{Cuiabá}

**Tabela 2.4** Dados Simbólicos.

$w_u$	$Y_1$	$Y_3$	$Y_4$	$Y_6$	$Y_7$	$n_u$
$w_1$	{N,NE,SE,S}	{S, N}	{M(5/11),F(6/11)}	[72.0,240.0]	[31.0,66.0]	11
$w_2$	{N,NE,SE,S}	{S, N}	{M(4/10),F(6/10)}	[36.0,120.0]	[24.0,43.0]	10
$w_3$	{N,NE,SE,S}	{S, N}	{M(4/7),F(3/7)}	[72.0,240.0]	[29.0,64.0]	7
$w_4$	{N,NE,S}	{S}	{M(4/6),F(2/6)}	[8.4,24.0]	[25.0,67.0]	6
$w_5$	{N,NE,SE}	{S}	{M(2/5),F(3/5)}	[0.0,6.0]	[16.0,23.0]	5
$w_6$	{CO}	{S}	{M(1/1)}	[120.0,120.0]	[50.0,50.0]	1

Uma observação simbólica tipicamente representa o conjunto de indivíduos que satisfazem a descrição associada ao conceito simbólico, ou categoria,  $w_u$  (Diday e Billard, 2006). Quando o número de indivíduos representados pelo conceito  $w_u$  é conhecido, ele é representado por  $n_u$ .

Um valor clássico da variável aleatória  $Y_j$ ,  $j = 1, \dots, p$  do indivíduo  $i = 1, \dots, n$ ,  $i \in \Omega$ ,  $\Omega = \{1, \dots, n\}$ , será denotado como  $x_{ij}$  e um valor simbólico será denotado  $\xi_{ij}$ . Ou seja,  $Y_j(i) = x_{ij}$  quando se tratar de uma variável clássica e  $Y_j(i) = \xi_{ij}$  quando se tratar de uma variável simbólica. Um valor clássico é um caso particular de um valor simbólico correspondente. Variáveis clássicas podem ser qualitativas (e.g.,  $X_{ij} =$

*Recife*) ou quantitativas (e.g.,  $X_{ij} = 24$ ). Variáveis simbólicas podem ser multivaloradas, intervalares ou modais com ou sem lógica, taxonomia e regras de dependência hierárquica.

### 2.1.1 Variáveis Multivaloradas

O valor de uma variável aleatória simbólica multivalorada  $Y$  é composto por um ou mais valores categóricos ou quantitativos retirados do domínio da variável. A quantidade de valores possíveis presentes nesse domínio é finita.

No conjunto de dados de censo demográfico presente na Tabela 2.3, os valores da variável  $Y_2 = Cidade$  para a categoria Médico ( $u = 1$ ) são:

$$Y_2 = \{\text{Recife, Manaus, Fortaleza, Vitória, São Paulo, Belém, Porto Alegre}\}.$$

Ou seja, os indivíduos que fazem parte dessa categoria moram em alguma das cidades listadas. Da mesma forma, os valores da variável  $Y_3 = Brasileiro$  para a mesma categoria, segundo a Tabela 2.4 são:

$$Y_3 = \{S, N\},$$

o que significa que alguns desses indivíduos são brasileiros e outros não.

Um caso particular do valor de uma variável simbólica multivalorada é uma observação qualitativa clássica, onde a lista de valores que compõe o valor de  $\xi$  tem apenas um valor. Por exemplo, o valor da variável  $Y_3 = Brasileiro$  para a categoria Estudante ( $u = 5$ ) é  $Y_3 = \{S\}$ , ou seja, todos os estudantes são brasileiros.

### 2.1.2 Variáveis Intervalares

Uma variável aleatória simbólica intervalar  $Y$ , como o próprio nome diz, toma valores em um intervalo,  $Y = \xi = [a, b] \subset \mathbb{R}^1$ , onde  $a \leq b, a, b \in \mathbb{R}^1$ . Quando os valores intervalares surgem da agregação de um conjunto de dados clássicos como na Tabela 2.2, os valores para  $a_{uj}$  e  $b_{uj}$  a variável simbólica  $j$  são achados da seguinte maneira:

$$\xi_{uj} = [\min_{i \in \Omega_u} x_{ij}, \max_{i \in \Omega_u} x_{ij}],$$

onde  $\Omega_u$  é o conjunto de  $i \in \Omega$  valores que constituem a categoria  $w_u$ .

No conjunto de dados de censo demográfico presente na Tabela 2.4, os valores da variável  $Y_6 = Renda\ anual$  para a categoria Porteiro ( $u = 4$ ) são  $Y_6 = [8.4, 24]$ , caracterizando a variável Renda anual como uma variável simbólica intervalar.

Uma variável quantitativa clássica é um caso particular da variável simbólica do tipo intervalo. Esse caso ocorre com as variáveis  $Y_6 = Renda\ anual$ , e  $Y_7 = Idade$ , para a categoria Agricultor ( $u = 6$ ). Seus respectivos valores são  $[120, 120]$  e  $[50, 50]$ . Esses

intervalos são chamados de intervalos degenerados e equivalem a um ponto em  $\mathbb{R}^1$ , ou seja, equivalem a um valor quantitativo clássico.

### 2.1.3 Variáveis Modais

Suponha que uma variável  $Y$  tome seus valores do domínio  $\{\eta_k; k = 1, 2, \dots\}$ . Então uma variável é modal se ela toma a forma

$$Y(w_u) = \xi_u = \{\eta_k, \pi_k; k = 1, \dots, s_u\}$$

para uma observação  $u$ , onde  $\pi_k$  é uma medida não-negativa associada a  $\eta_k$  e  $s_u$  representa o número de valores do domínio utilizados. Os valores  $\eta_k$  podem ser categóricos ou qualitativos e finitos ou infinitos. As medidas  $\pi_k$  podem ser pesos, probabilidades, frequências relativas, entre outros tipos, que correspondem ao componente  $\eta_k$ .

No conjunto de dados de censo demográfico presente na Tabela 2.4, os valores da variável  $Y_4 = \text{Sexo}$  para a categoria Engenheiro ( $u = 2$ ) são:

$$Y_4 = \{M(4/10), F(6/10)\},$$

ou seja, 40% dos engenheiros são homens e 60% dos engenheiros são mulheres.

Variáveis modais incluem também distribuições, modelos e uma variedade de outras entidades. Um caso particular das variáveis modais são as variáveis do tipo histograma. Suponha que uma variável  $Y$  tome seus valores de um número finito de intervalos não sobrepostos  $\{[a_k, b_k), k = 1, 2, \dots\}$  onde  $a_k \leq b_k$ . Então uma variável aleatória simbólica do tipo histograma toma a forma

$$Y(w_u) = \xi_u = \{[a_k, b_k), p_{uk}; k = 1, \dots, s_u\},$$

onde  $s_u < \infty$  é o número finito de intervalos que formam o valor de  $Y(w_u)$  para a observação  $w_u$  e  $p_{uk}$  é o peso para o intervalo  $[a_k, b_k)$ ,  $k = 1, \dots, s_u$ , com  $\sum_{k=1}^{s_u} p_{uk} = 1$ .

No conjunto de dados de censo demográfico presente na Tabela 2.4, a variável intervalar  $Y_6 = \text{Renda anual}$  pode ser também representada da seguinte maneira para a categoria Engenheiro ( $u = 2$ ):

$$Y_6 = \{[20, 40), 0.3; [40, 60), 0.3; [60, 80), 0.0; [80, 100), 0.3; [100, 120], 0.1\},$$

ou seja, 30% dos engenheiros ganham entre 20000 e 40000 reais, 30% ganham entre 40000 e 60000 reais, nenhum deles ganha entre 60000 e 80000 reais, 30% ganham entre 80000 e 100000 reais e 10% ganham entre 100000 e 120000 reais.

# 3

## Aprendizagem Baseada em Protótipos

*Aprender e não recordar o que se estudou,  
equivale a semear e não colher.*

—TEXTOS JUDAICOS

Este trabalho baseia-se em um algoritmo da família LVQ e essa família consiste de algoritmos de Aprendizagem Supervisionada baseada em protótipos. Portanto, é interessante analisar brevemente o que já está consolidado na literatura desse campo (especialmente no que diz respeito a dados simbólicos), a fim de situar e contextualizar melhor o método proposto adiante.

O objetivo da Aprendizagem Supervisionada (AS) é construir um modelo da distribuição das classes a partir de características de predição, fornecidas pelos exemplos extraídos de um conjunto de treinamento. O classificador resultante passa a ser usado para designar rótulos de classes para instâncias de teste cuja classe é desconhecida, embora as características de predição sejam conhecidas (Aly, 2005). Alguns dos algoritmos de AS baseados em protótipos são: k-Vizinhos Mais Próximos e classificadores baseados em k-Médias.

Na AS baseada em protótipos, um subconjunto de protótipos é escolhido no espaço de dados de entrada para representar as regiões de classes. O processo de treinamento de algoritmos baseados em protótipos consiste em tentar posicionar os protótipos de forma ótima, ou seja, que aproxime a melhor representação das classes. Para classificação, identifica-se o protótipo mais relacionado ao dado de exemplo e assume-se que a classe

do dado é a mesma do protótipo.

### 3.1 k-Vizinhos mais Próximos

O k-Vizinhos mais Próximos, do inglês k-Nearest Neighbors (k-NN), é um dos mais antigos algoritmos de classificação (Aly, 2005). Sua implementação é bastante simples: para classificar um exemplo, calcula-se sua distância para todas as instâncias de treinamento, para achar os  $k$  vizinhos mais próximos. A classe do exemplo é definida como a classe mais freqüente entre os  $k$  vizinhos mais próximos.

Essa forma de classificação pode trazer problemas. Por exemplo, suponha que na situação demonstrada na Figura 3.1, o parâmetro  $k$  tem valor 3 e que a classe verdadeira do exemplo representado pelo círculo indicado pela interrogação seja a classe azul. Com esse valor de  $k$  o algoritmo iria identificar como vizinhos mais próximos 3 círculos vermelhos, ou seja, a classificação seria incorreta. Portanto é preciso achar um valor de  $k$  que evite erros. Valores altos de  $k$  geralmente reduzem os efeitos do ruído nos dados, mas tornam as fronteiras entre as classes menos distintas. Além disso, não se escolhe valores pares para  $k$ , pois isso poderia levar a empates na escolha da classe.

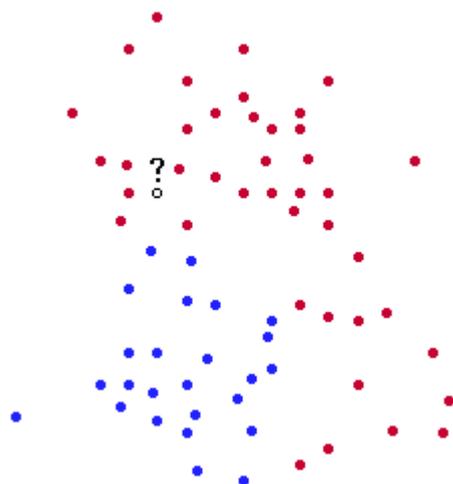
Uma adaptação que é feita ao k-NN básico para evitar o problema relatado acima é a ponderação das distâncias, com isso, elementos mais distantes influenciam menos na escolha da classe do exemplo. O peso para cada elemento pode assumir um valor  $1/d$  em que  $d$  é o valor da distância do elemento ao padrão de exemplo. A idéia de vizinho mais próximo está por trás dos algoritmos LVQ, pois, no seu processo de treinamento, procura-se pelo protótipo mais próximo ao exemplo.

### 3.2 Versões otimizadas do k-NN

O k-NN tradicional pode ser custoso por comparar o objeto de teste com todos os objetos de treinamento para achar os  $k$  vizinhos mais próximos. Por isso, algumas versões do k-NN surgiram. Essas versões tem como princípio a escolha de representantes das classes entre os objetos de treinamento. Esses objetos são chamados de protótipos.

#### 3.2.1 k-Vizinhos Mais Próximos Editados

O algoritmo k-Vizinhos Mais Próximos Editados, do inglês Edited Nearest Neighbors (ENN), proposto por Wilson (1972), foi projetado para ser um redutor de ruídos na região de fronteira das classes, no entanto tem pouco poder de redução. Pode ser usado em



**Figura 3.1** Exemplo de k-NN, classe do novo elemento será definida como a classe mais comum entre os  $k$  vizinhos mais próximos.

pré-processamento de dados. Seus passos são os seguintes:

1. Para cada objeto  $x_q$  do conjunto de dados ache seus  $k$  vizinhos mais próximos, utilizando a medida de dissimilaridade escolhida.
2. Elimine do conjunto de dados os objetos  $x_i$ , onde  $i = 1, \dots, k$  e  $C(x_i) \neq C(x_q)$ .

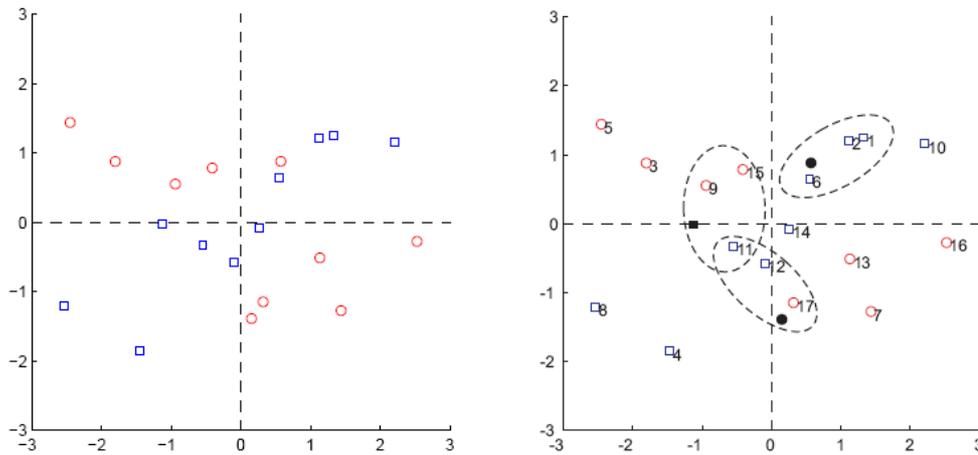
A figura Figura 3.2 apresenta um exemplo de ENN com  $k = 3$ . Dois objetos da classe círculo e um objeto da classe quadrado estão preenchidos de preto e são os objetos de teste. O ENN vai eliminar do conjunto de dados os objetos que estiverem entre os  $k$  vizinhos mais próximos dos objetos de teste e forem de classes diferentes. O resultado do processo aparece na Figura 3.3.

### 3.2.2 k-Vizinhos Mais Próximos Condensados

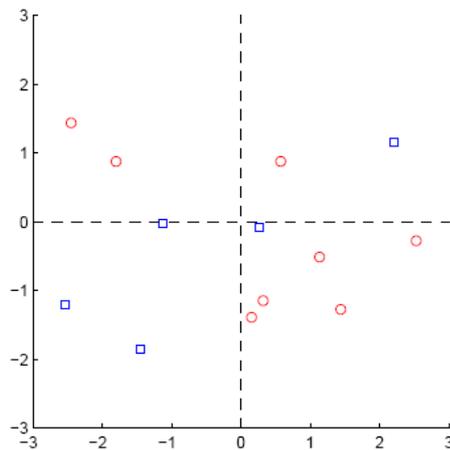
O algoritmo k-Vizinhos Mais Próximos Condensados, do inglês Condensed Nearest Neighbors (CNN), proposto por Hart (1968), visa à eliminação de informação redundante e sua ideia é encontrar o menor subconjunto  $S$  do conjunto de dados  $E$  que classifica corretamente todos os elementos de  $E$  usando  $k = 1$ . Os passos do algoritmo são os seguintes:

1. Inicialize  $S$  com um objeto de cada classe escolhido aleatoriamente.
2. A seguir, utilize o primeiro vizinho mais próximo ( $k = 1$ ) dos objetos de  $S$  para classificar todos os objetos de  $E$ .

### 3.2. VERSÕES OTIMIZADAS DO K-NN



**Figura 3.2** Exemplo de ENN.



**Figura 3.3** Resultado do ENN.

- (a) Aqueles que forem classificados corretamente são descartados.
- (b) Os objetos de  $E$  que forem classificados incorretamente são incorporados ao conjunto  $S$ .

O resultado do processo é a eliminação de objetos mais afastados da fronteira de classificação, por isso o nome k-Vizinhos Mais Próximos Condensados. A obtenção do menor conjunto que classifica corretamente os objetos de  $E$  não é garantida, pois é afetada pela ordem de seleção dos objetos.

### 3.2.3 Links de Tomek

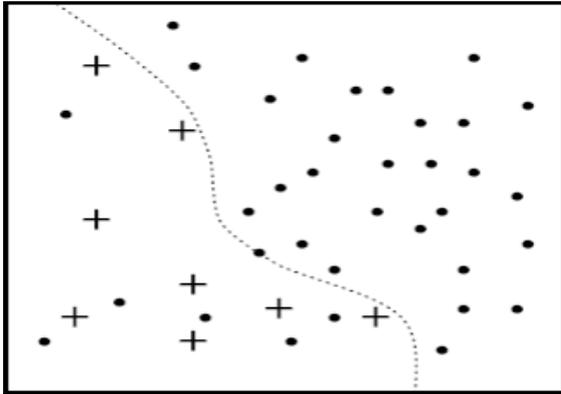
A ideia desse algoritmo proposto por Tomek (1976) é eliminar ruído, ou fortes candidatos a ruído, nas regiões de fronteiras. Um link de Tomek é definido como um par de objetos  $x_{q1}$  e  $x_{q2}$  de classes distintas, posicionados de forma que não exista nenhum objeto  $x_{q3}$  tal que  $d(x_{q3}, x_{q1}) < d(x_{q1}, x_{q2})$  ou  $d(x_{q3}, x_{q2}) < d(x_{q1}, x_{q2})$ .

A eliminação pode ser feita de duas formas e servir dois propósitos: para resolver desbalanceamento entre classes, apenas objetos da classe majoritária que formam links de Tomek são removidos; para limpeza de dados, ambos os objetos dos links de Tomek são removidos. A remoção dos objetos de regiões de dúvida causa o aparente afastamento das classes.

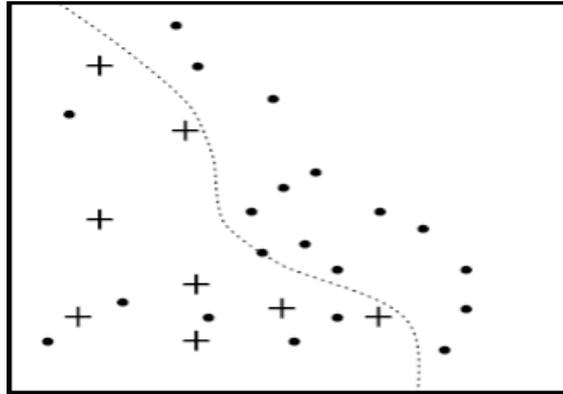
### 3.2.4 Seleção Unilateral

O método k-Vizinhos com Seleção Unilateral, do inglês One-Sided Selection (OSS), consiste na aplicação do algoritmo CNN seguida da eliminação dos links de Tomek do conjunto de dados, com o intuito de resolver problemas de desbalanceamento entre classes (Kubat e Matwin, 1997). A estratégia é simples: a utilização do CNN elimina os objetos distantes da fronteira de classificação, diminuindo o desbalanceamento entre as classes e a eliminação dos links de Tomek remove o ruído entre as classes, diminuindo as classificações errôneas.

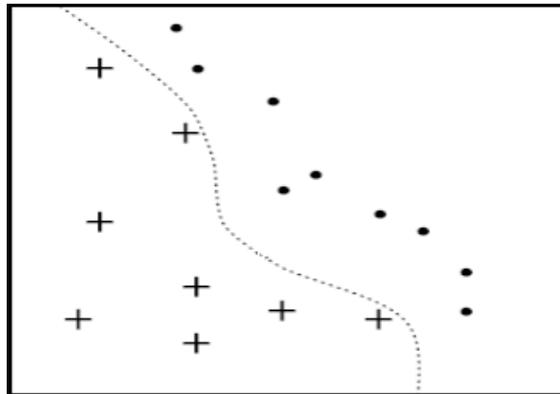
A Figura 3.4 apresenta um conjunto de dados com ruído e desbalanceamento entre as classes círculo e cruz. A Figura 3.5 mostra o conjunto após a aplicação do CNN, com eliminação dos objetos que não se situam próximos à zona de fronteira. Por fim, a Figura 3.6 mostra o conjunto de dados final, após a eliminação dos links de Tomek, finalizando o algoritmo OSS.



**Figura 3.4** Conjunto de dados original com ruído e desbalanceamento entre as classes.



**Figura 3.5** Conjunto após a aplicação do CNN, ainda com ruído entre as classes.



**Figura 3.6** Resultado do OSS, após a eliminação dos links de Tomek.

### 3.3 k-NN Simbólico

Appice *et al.* (2006) introduziram uma abordagem de lazy learning, chamada Optimal Local Distance-Weighted Symbolic k-NN (OLD-SKNN), que estende um k-vizinhos mais próximos ponderado por distância clássica para dados intervalares e modais. O OLD-SKNN utiliza validação cruzada para escolher automaticamente o valor de  $k$  e dá como resultado uma variável simbólica modal, contendo a lista de probabilidades das classes para o elemento de entrada. O algoritmo é descrito da seguinte forma:

1. Dado um objeto de teste  $x_q$  para ser classificado de acordo com a medida de dissimilaridade  $d$ , sejam  $x_1, x_2, \dots, x_k$  os  $k$  objetos de treinamento mais parecidos com o objeto de teste, ou seja os  $k$  objetos para os quais  $d(x_q, x_i)$  é mínimo.

2. Seja  $n$  o número de classes. É possível distinguir três casos possíveis:

(a) Para todos os vizinhos mais próximos  $x_i$  tais que  $d(x_q, x_i) = 0$  (idênticos ao objeto de teste), cada classe  $C(x_i)$  é igual a uma mesma classe  $c$ . Com isso, as probabilidades de classe resultantes são:

$$P(C'(x_q) = c) = 1 \text{ e } P(C'(x_q) = v_j) = 0 \quad \forall j = 1, \dots, n \text{ tal que } v_j \neq c.$$

(b) As classes  $C(x_i)$  de todos os vizinhos mais próximos  $x_i$  tais que  $d(x_q, x_i) = 0$  podem ser diferentes. Nesse caso, as probabilidades de classes são calculadas de acordo com o número de vizinhos mais próximos idênticos pertencentes a cada classe:

$$P(C'(x_q) = v_j) = \frac{\#x_i : C'(x_i) = v_j}{k} \quad \forall j = 1, \dots, n.$$

(c) Para todos os vizinhos mais próximos  $x_i$ ,  $d(x_q, x_i) \neq 0$  para  $i = 1, \dots, k$ . Seja  $\omega_i = 1/d(x_q, x_i)$  o peso associado ao objeto de treinamento  $x_i$ . Defina-se  $\Omega_j = \sum_{i=1}^k \omega_i \delta(v_j, C(x_i))$  para  $j = 1, \dots, n$ , onde  $\delta(v_j, C(x_i)) = 1$  se  $C(x_i) = v_j$ ; caso contrário  $\delta(v_j, C(x_i)) = 0$ . Com isso, as probabilidades de classe resultantes são:

$$P(C'(x_q) = v_j) = \frac{\frac{\#x_i : C'(x_i) = v_j}{k} \Omega_j}{\sum_{j=1}^n \frac{\#x_i : C'(x_i) = v_j}{k} \Omega_j} \quad \forall j = 1, \dots, n$$

### 3.4 Classificador baseado em k-Médias

O algoritmo k-Médias (em inglês k-Means) original é um algoritmo de clustering não-supervisionado, que define partições e centros de partições em um conjunto de dados sem rótulos de classes. O  $k$  é um parâmetro que representa o número de centros de partições que o algoritmo deve encontrar e o algoritmo move os centros com o intuito de minimizar a variação interna das partições. Assim, dado um conjunto inicial de centros (tipicamente escolhidos aleatoriamente entre as observações do conjunto de treinamento), o algoritmo k-Médias alterna os seguintes passos até sua convergência:

1. para cada centro, identificar o subconjunto de pontos que são mais próximos a ele do que a qualquer outro centro;
2. computa-se a média de cada atributo para os pontos em cada partição e esse vetor de médias torna-se o novo centro daquela partição.

Uma alteração interessante do k-Médias é a adaptação do algoritmo para torná-lo um classificador supervisionado Hastie *et al.* (2001). Com isso, os passos do novo algoritmo são três:

1. aplicar o k-Médias particional para os dados de cada classe separada, usando  $R$  protótipos por classe;
2. designar um rótulo de classe para cada um dos protótipos;
3. classificar um novo padrão  $x$  para a classe do protótipo mais próximo.

#### 3.4.1 c-Médias Difuso Intervalar

Métodos c-Médias difusos são baseados no k-Médias, mas permitem que um objeto pertença a dois ou mais clusters. Para agrupar dados simbólicos intervalares, de Carvalho (2007) propôs métodos c-Médias difusos adaptativos e não-adaptativos. Esses métodos geram partições difusas dos dados ao otimizar um critério de adequação baseado em distâncias Euclidianas quadráticas entre vetores de intervalos. Assim como o k-Médias clássico, esse algoritmo também pode ser utilizado no contexto de classificação supervisionada.

# 4

## O LVQ

*É fazendo que se aprende a fazer aquilo que se  
deve aprender a fazer.*

—ARISTÓTELES

Como já foi citado nesse trabalho, LVQ se refere a toda uma família de algoritmos, como LVQ1, LVQ2, LVQ3 e OLVQ1. O LVQ foi proposto por Kohonen, que também introduziu os Self-organizing Maps (SOM). LVQ e SOM, são relacionados, pois são baseados em protótipos e seus processos de convergência consistem em fazer os protótipos se distribuírem pelo espaço de dados, a fim de representar melhor as regiões de classes (Kohonen, 2001). No entanto, o LVQ é um algoritmo de aprendizado supervisionado e o SOM é não-supervisionado. Além disso, o algoritmo SOM define vizinhanças ao redor do neurônio "vencedor", de forma que os neurônios da vizinhança também são afetados em certo grau. Esse conceito não é utilizado no LVQ básico.

### 4.1 Decisão Ótima

O problema da decisão ótima ou reconhecimento estatístico de padrões consiste tradicionalmente em aproximar a função que indica a probabilidade de uma amostra  $x$  pertencer a determinada classe  $C$ , com a mínima taxa de erros de classificação. Ao achar essa função,  $f(x)$ , admite-se que  $x$  pertence à classe que tem o valor máximo de  $f(x)$ .

O LVQ, no entanto, é baseado numa filosofia completamente diferente. Supondo que existem  $C_1, \dots, C_K$  classes. Seja  $\Omega = \{(\mathbf{x}_i, y_i)\} (i = 1, \dots, n)$  um conjunto de treinamento. Cada padrão  $i$  é descrito por um vetor de  $p$  atributos quantitativos  $\mathbf{x}_i = (x_i^1, \dots, x_i^p)$  e um atributo discreto de classe  $Y$  que toma valores no conjunto discreto  $1, \dots, K$ . Seja  $\{(\mathbf{w}_m, y_m)\} (m = 1, \dots, M)$  um conjunto de protótipos, cada um representado por um vetor

de  $p$  atributos quantitativos  $\mathbf{w}_m = (w_m^1, \dots, w_m^p)$  e um atributo discreto de classe  $Y$  que toma valores no conjunto discreto  $1, \dots, K$ .

Assume-se que, para cada classe  $k(k = 1, \dots, K)$ , é designado um subconjunto de protótipos, de forma que o número total de protótipos é  $M$ . Então, para cada padrão de treinamento  $i(\mathbf{x}_i, y_i)$ , encontra-se o protótipo  $m(\mathbf{w}_m, y_m)$  que tem a menor distância Euclidiana em relação a  $i$ .

## 4.2 O LVQ1

Supondo que um certo número de protótipos é designado para cada classe  $k$ , pela regra do 1-vizinho mais próximo, determina-se que o padrão  $i$  extraído de  $\Omega$  pertence à mesma classe à qual pertence o protótipo  $m$  mais próximo a  $i$ . Assim,

$$c = \arg \min_m \left\{ d_E(\mathbf{x}_i, \mathbf{w}_m) = \sum_{j=1}^p (x_i^j - w_m^j)^2 \right\} \quad (4.1)$$

define o índice do protótipo  $m$  mais próximo a  $i$  de acordo com a distância Euclidiana  $d_E$ . Logo,  $c$ , o índice do "vencedor", depende do padrão  $i$  e de todos os  $M$  protótipos.

Seja  $\mathbf{x}_i$  o vetor das variáveis da instância de dados  $i$  e  $\mathbf{w}_m(t)$  os valores sequenciais do vetor das variáveis do protótipo  $m$  nos instantes  $t = 0, 1, 2, \dots$ . Tais valores podem ser achados, de forma a minimizar a taxa de erros de classificação, segundo o processo de aprendizagem a seguir.

### 1. Inicialização

1.1 Seleciona-se aleatoriamente  $M$  padrões distintos do conjunto de treinamento  $\Omega$ , produzindo o subconjunto inicial de protótipos  $\{(\mathbf{w}_1(t), y_1), \dots, (\mathbf{w}_M(t), y_M)\}$ .

1.2 Inicie a taxa de aprendizagem  $\alpha(t)$  com um valor preferencialmente pequeno, como 0.1.

2. **Passo de atualização dos protótipos:** Escolha um padrão de treinamento  $i(\mathbf{x}_i, y_i)$  de  $\Omega$  aleatoriamente com reposição.

2.1 Defina o protótipo vencedor  $c$  tal que

$$c = \arg \min_{m=1, \dots, M} d_E(\mathbf{x}_i, \mathbf{w}_m)$$

2.2 Se classe( $i$ ) = classe( $c$ ) faça  $\mathbf{w}_c(t+1) = \mathbf{w}_c(t) + \alpha(t)[\mathbf{x}_i - \mathbf{w}_c(t)]$ .

2.3 Se  $\text{classe}(i) \neq \text{classe}(c)$  faça  $\mathbf{w}_c(t+1) = \mathbf{w}_c(t) - \alpha(t)[\mathbf{x}_i - \mathbf{w}_c(t)]$ .

3. **Atualização da taxa de aprendizagem:** A taxa de aprendizagem deve decrescer até zero com cada iteração. Repetir do passo 2 até convergir.

A taxa de aprendizagem  $\alpha(t)$  decresce, normalmente, monotonicamente com o tempo. Recomenda-se que  $\alpha$  comece pequena, por exemplo 0.1. A fórmula exata para a atualização de  $\alpha(t)$  não é crucial (Kohonen, 2001), podendo até decrescer linearmente até zero, desde que o número de passos de treinamento seja suficiente.

### 4.3 O LVQ1 Otimizado

Nesta variação do LVQ1 básico (nesse trabalho chamada de *OLVQ*), uma taxa de aprendizagem individual  $\alpha_m(t)$  é designada para cada protótipo  $m$ , de forma que cada protótipo possa convergir independentemente e modelar melhor os dados representados por ele (Kohonen, 2001). Assim, o seguinte processo de aprendizagem é obtido:

#### 1. Inicialização

- 1.1 Seleciona-se aleatoriamente  $M$  padrões distintos do conjunto de treinamento  $\Omega$ , produzindo o subconjunto inicial de protótipos  $\{(\mathbf{w}_1(t), y_1), \dots, (\mathbf{w}_M(t), y_M)\}$ .
- 1.2 De  $m = 1$  até  $M$  inicie  $\alpha_m(t)$ .

2. **Passo de atualização dos protótipos:** Escolha um padrão de treinamento  $i$  de  $\Omega$  aleatoriamente com reposição.

- 2.1 Defina o protótipo vencedor  $c$  tal que

$$c = \arg \min_{m=1, \dots, M} d_E(\mathbf{x}_i, \mathbf{w}_m)$$

- 2.2 Se  $\text{classe}(i) = \text{classe}(c)$  faça  $\mathbf{w}_c(t+1) = \mathbf{w}_c(t) + \alpha_c(t)[\mathbf{x}_i - \mathbf{w}_c(t)]$ .

- 2.3 Se  $\text{classe}(i) \neq \text{classe}(c)$  faça  $\mathbf{w}_c(t+1) = \mathbf{w}_c(t) - \alpha_c(t)[\mathbf{x}_i - \mathbf{w}_c(t)]$ .

- 2.4 Atualize

$$\alpha_c(t+1) = \frac{\alpha_c(t)}{1 + s(t)\alpha_c(t)}$$

onde  $s(t) = +1$  se  $i$  e  $c$  pertencem à mesma classe, e  $s(t) = -1$  se  $i$  e  $c$  pertencem a classes diferentes.

3. **Critério de parada:** Se todos os  $\alpha_m(t)$  ( $m = 1, \dots, M$ ) se aproximarem arbitrariamente de 0, então o algoritmo convergiu. Caso contrário, repetir o passo 2.

É preciso ter precaução quanto a essa forma de atualização das taxas de aprendizagem, pois  $\alpha_c(t)$  pode também aumentar. Deve-se evitar que  $\alpha_c(t)$  torne-se maior do que 1, o que pode ser imposto no próprio algoritmo. Para valores iniciais de  $\alpha_m$  pode-se escolher valores maiores do que no LVQ1, pois no OLVQ as taxas individuais podem decrescer mais rápido (Kohonen, 2001). Heuristicamente observa-se que valores entre 0.3 e 0.5 apresentam bons resultados. O OLVQ foi o algoritmo escolhido como base para o método intervalar com distâncias adaptativas apresentado nessa dissertação.

## 4.4 Outras variações

Existem outras versões do LVQ básico. Em geral, essas modificações não alteram a forma como o algoritmo toma as decisões na classificação. São alterações na etapa de aprendizagem. Existem versões que atualizam mais de um protótipo a cada iteração, como o LVQ2 e o LVQ3; versões batch, como o LVQ1-Batch; entre outras.

### 4.4.1 O LVQ2

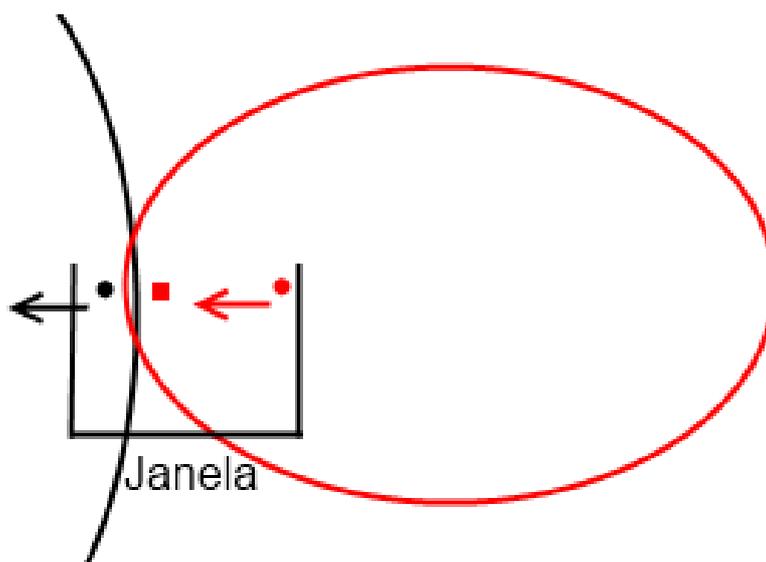
Nessa variação do LVQ básico, dois protótipos que são os vizinhos mais próximos, segundo a distância Euclidiana, de um padrão  $i$  ( $\mathbf{x}_i, y_i$ ), retirado aleatoriamente do conjunto de treinamento  $\Omega$ , são atualizados simultaneamente. A ideia é mover o plano central dos dois protótipos para a região de fronteira das classes. Se o protótipo mais próximo  $a$  ( $\mathbf{w}_a, y_a$ ) pertencer à classe errada e o segundo mais próximo  $b$  ( $\mathbf{w}_b, y_b$ ) pertencer à classe correta,  $\mathbf{x}_i$  deve cair numa zona de valores chamada de “janela”.

A Figura 4.1 exemplifica o LVQ2. Existem duas classes: a preta e a vermelha. Os protótipos mais próximos ao dado de treinamento (representado pelo quadrado vermelho) são o protótipo representado pelo círculo preto (portanto, da classe preta) e o protótipo representado pelo círculo vermelho (portanto, da classe vermelha), nessa ordem. Se suas distâncias em relação ao dado de treinamento satisfizerem a “janela”, eles serão ajustados e o seu plano central será deslocado em direção à região de fronteira das classes.

Sejam  $d_a$  e  $d_b$  as distâncias Euclidianas entre  $\mathbf{x}_i$  e  $\mathbf{w}_a$  e entre  $\mathbf{x}_i$  e  $\mathbf{w}_b$ , respectivamente. Então,  $\mathbf{x}_i$  cairá numa “janela” de largura  $W$  se

$$\min\left(\frac{d_a}{d_b}, \frac{d_b}{d_a}\right) > s, \text{ onde } s = \frac{1 - W}{1 + W} \quad (4.2)$$

Essa janela não deve ser muito “estreita”, ou seja, o valor de  $W$  não deve ser muito pequeno, pois isso faria o termo  $\frac{1-W}{1+W}$  se tornar maior, reduzindo o número de atualizações



**Figura 4.1** Exemplo de LVQ2.

de protótipos. Recomenda-se uma largura de “janela”  $W$  de 0.2 a 0.3 (Kohonen, 2001).

#### 4.4.2 O LVQ3

O LVQ2 não leva em conta o que pode acontecer com a localização dos protótipos a longo prazo. Além disso, no LVQ2 o protótipo  $b$  da classe correta é atualizado com uma magnitude maior do que o protótipo  $a$  da classe errada, pois o protótipo  $b$  é o segundo mais próximo, portanto o termo  $[x_i(t) - w_b(t)]$  é maior do que o termo  $[x_i(t) - w_a(t)]$ . Assim, parece necessário introduzir correções que garantam que os protótipos continuem aproximando as distribuições das classes (Kohonen, 2001).

A diferença do LVQ3 para o LVQ2 é que o LVQ3 permite que os dois protótipos escolhidos sejam da mesma classe. Quando  $i$ ,  $a$  e  $b$  pertencem à mesma classe, faz-se:

$$w_g(t+1) = w_g(t) + \varepsilon \alpha(t) [x_i(t) - w_g(t)], \text{ para } g \in a, b \quad (4.3)$$

Esse parâmetro  $\varepsilon$  é usado apenas quando  $i$ ,  $a$  e  $b$  pertencem à mesma classe e serve para suavizar a atualização de protótipos que já estejam bem posicionados.

Numa série de experimentos, foram achados valores aplicáveis de  $\varepsilon$  entre 0.1 e 0.5, relacionados a  $W = 0.2$  ou 0.3. O valor ótimo de  $\varepsilon$  demonstra ter uma relação direta com o tamanho  $W$  da "janela", devendo ser menor para janelas mais estreitas. Com o algoritmo descrito, o posicionamento ótimo dos protótipos parece não mudar com a continuação do processo de aprendizagem (Kohonen, 2001).

### 4.4.3 O LVQ1-Batch

O LVQ1 básico pode ser expresso numa forma batch como segue:

1. Para se iniciar os protótipos, pode-se utilizar um processo não-supervisionado, como o *SOM*, precedente, em que a classificação de  $i$  ainda não é levada em consideração.
2. Os  $i$  são apresentados de novo, desta vez listando-os, junto com seus atributos de classe sob os protótipos vencedores correspondentes.
3. Determina-se as classes dos protótipos de acordo com a classe mais freqüente nas suas listas de padrões.
4. Para cada protótipo  $m$ , tome para seu novo valor do vetor de código a entidade

$$w_m^* = \frac{\sum_{t'} s(t')x(t')}{\sum_{t'} s(t')} \quad (4.4)$$

onde os  $x(t')$  representam os valores dos padrões listados sob o protótipo  $m$  e os  $s(t')$  indicam se os  $x(t')$  pertencem à mesma classe do protótipo  $m$  ou não.

5. Repita a partir de 2 algumas vezes.

Para manter a estabilidade, pode ser necessário checar o valor de  $\sum_{t'} s(t')$ , pois, se for negativo, não se deve atualizar o protótipo. Um fator importante é que, ao contrário do LVQ1 básico, o LVQ1-Batch permite a mudança dinâmica da classe dos protótipos.

## 4.5 Versões Adaptativas

Algumas versões adaptativas foram propostas como parte de esforços para modelar melhor diversas situações presentes em conjuntos de dados, como a influência das variáveis na distribuição dos dados de um conjunto ou de uma classe específica ou a formação de subregiões dentro das classes. Esses algoritmos são chamados de adaptativos porque buscam se adaptar aos dados para melhor representá-los.

Silva Filho e Souza (2011) introduziram o Optimized Learning Vector Quantization with an Adaptive Euclidean Distance by Class (OLVQ-AEDC) e o Optimized Learning Vector Quantization with an Adaptive Euclidean Distance by Prototype (OLVQ-AEDP), para modelar conjuntos de dados constituídos de classes ou subregiões de classes (representadas pelos protótipos) que possuem formas diferentes.

### 4.5.1 OLVQ-AEDC

Uma abordagem para modelar melhor classes com formatos variados é buscar capturar a influência das variáveis na distribuição dos dados das classes. Para isso, pode-se manter um vetor de pesos  $\boldsymbol{\lambda}_k = (\lambda_k^1, \dots, \lambda_k^p)$  que tem pesos para cada uma das  $p$  variáveis, para cada classe  $k$ .

Uma distância com essa característica deve respeitar um critério  $J$  baseado na regra do vizinho mais próximo definido como:

$$J = \sum_{k=1}^K \sum_{i \in k} d_k(\mathbf{x}_i, \mathbf{w}_c(t)) \delta_{ci} \quad (4.5)$$

onde  $d_k$  é a distância Euclidiana adaptativa da classe  $k$  entre um padrão  $i$  e um protótipo  $c(t)$  da classe  $k$  na iteração  $t$ , tal que  $c$  é o índice do protótipo mais próximo a  $i$  e  $\delta_{ci} = 1$  se  $i$  e  $c$  pertencem à mesma classe e  $\delta_{ci} = 0$ , caso contrário.

O critério  $J$  acima representa a soma das distâncias entre padrões  $i$  e seus protótipos vencedores  $c$  que pertencem à mesma classe  $k$ . A distância Euclidiana adaptativa da classe  $k$  ( $d_k$ ) é calculada como segue:

$$d_k(\mathbf{x}_i, \mathbf{w}_c(t)) = \sum_{j=1}^p \lambda_k^j (x_i^j(t) - w_c^j(t))^2 \quad (4.6)$$

A adaptabilidade de  $d_k$  é expressa pelo vetor de pesos  $\boldsymbol{\lambda}_k = (\lambda_k^1, \dots, \lambda_k^p)$ , que representa a informação da dispersão da classe. Esses vetores de pesos  $\boldsymbol{\lambda}_k$ , que minimizam o critério da equação (4.5), seguindo as restrições  $\lambda_k^j > 0$  e  $\prod_{j=1}^p \lambda_k^j = 1$ , são atualizados de acordo com a seguinte expressão:

$$\lambda_k^j = \frac{\{\prod_{h=1}^p (\sum_{i \in C_k} (x_i^h - w_c^h)^2)\}^{\frac{1}{p}}}{\sum_{i \in C_k} (x_i^j - w_c^j)^2} \quad (4.7)$$

onde  $h = 1, \dots, p$  é o conjunto dos índices das variáveis,  $k = 1, \dots, K$  é o conjunto dos índices das classes,  $C_k$  é o conjunto de padrões corretamente classificados como pertencendo à classe  $k$  com seus protótipos vencedores  $c$ , de acordo com a distância  $d_k$ .

Então, para cada classe  $k$ , uma soma é calculada para cada variável  $j$

$$\Delta_k^j = \sum_{i \in C_k} (x_i^j - w_c^j)^2 \delta_{ci} \quad (4.8)$$

da diferença quadrática entre os valores de cada padrão corretamente classificado como pertencendo à classe  $k$  e os protótipos vencedores correspondentes.  $\delta_{ci} = 1$  se  $classe(i) =$

classe( $c$ ) e  $\delta_{ci} = 0$ , caso contrário. Com isso, a equação (4.7) pode ser reescrita como:

$$\lambda_k^j = \frac{\{\prod_{h=1}^p (\Delta_k^h)\}^{\frac{1}{p}}}{\Delta_k^j} \quad (4.9)$$

A distância  $d_k$  deve capturar as diferenças das formas das classes. Caso essas diferenças não sejam relevantes, a distância deve render resultados equivalentes aos da distância Euclidiana padrão.

### O algoritmo

O classificador OLVQ-AEDC é um algoritmo online, então a soma  $\Delta_k^j$  definida na equação (4.8) deve ser atualizada toda vez que um protótipo da classe  $k$  afeta um padrão da classe  $k$ . Após isso, o vetor de pesos  $\lambda_k$  deve ser recalculado.

O problema é que, com a atualização constante dos protótipos, a contribuição de cada protótipo para  $\Delta_k^j$  se torna desatualizada. Para resolver esse problema, a atualização de  $\Delta_k^j(t)$  é uma soma ponderada da nova diferença quadrática entre os padrões de treinamento e os seus protótipos vencedores  $c$  e o valor anterior  $\Delta_k^j(t-1)$ , usando a taxa de aprendizagem  $\alpha_c(t)$  do protótipo vencedor  $c(t)$ . Isso é expresso pela fórmula:

$$\Delta_k^j = ((1 - \alpha_c(t))\Delta_k^j(t-1)) + (\alpha_c(t)(x_i^j(t) - w_c^j(t))^2) \quad (4.10)$$

Os passos do algoritmo *OLVQ-AEDC* são os seguintes:

#### 1. Inicialização

- 1.1 Escolha  $M$  padrões distintos do conjunto de dados de treinamento  $\Omega$  aleatoriamente, gerando o subconjunto inicial de protótipos  $\{(\mathbf{w}_1(t), y_1), \dots, (\mathbf{w}_M(t), y_M)\}$  no instante  $t = 0$ .
- 1.2 De  $m = 1$  a  $M$  faça  $\alpha_m(t) = 0.3$ .
- 1.3 De  $k = 1$  a  $K$  e  $j = 1, \dots, p$  faça  $\lambda_k^j = 1$  e  $\Delta_k^j = 0$ .

#### 2. Escolha um padrão $i$ ( $\mathbf{x}_i, y_i$ ) de $\Omega$ aleatoriamente com reposição

##### • Passo de atualização do protótipo:

- 2.1 Defina o protótipo vencedor  $c$  tal que

$$c = \arg \min_{m=1, \dots, M} d_k(\mathbf{x}_i, \mathbf{w}_m)$$

2.2 Se classe(i) = classe(c) faça  $\mathbf{w}_c(t+1) = \mathbf{w}_c(t) + \alpha_c(t)[\mathbf{x}_i - \mathbf{w}_c(t)]$

2.2.1 Atualize as somas  $\Delta_k^j$  para  $j = 1, \dots, p$ , onde  $k$  é a classe de  $c$ , usando a equação (5.7)

2.2.2 Para  $j = 1, \dots, p$ , compute  $\lambda_k^j$  com a equação (4.9), onde  $k$  é a classe de  $c$

2.3 Se classe(i)  $\neq$  classe(c) faça  $\mathbf{w}_c(t+1) = \mathbf{w}_c(t) - \alpha_c(t)[\mathbf{x}_i - \mathbf{w}_c(t)]$

2.4 Atualize

$$\alpha_c(t+1) = \frac{\alpha_c(t)}{1 + s(t)\alpha_c(t)}$$

onde  $s(t) = +1$  se  $i$  and  $c$  pertencem à mesma classe e  $s(t) = -1$ , caso contrário

3. **Critério de parada:** Se, para todo  $\alpha_m(t) \leq 0.00005$  ( $m = 1, \dots, M$ ), então PARE, senão vá para o passo 2.

O valor 0.00005 foi verificado heurísticamente como um bom valor para indicar que  $\alpha_m(t)$  se aproximou bastante de 0.

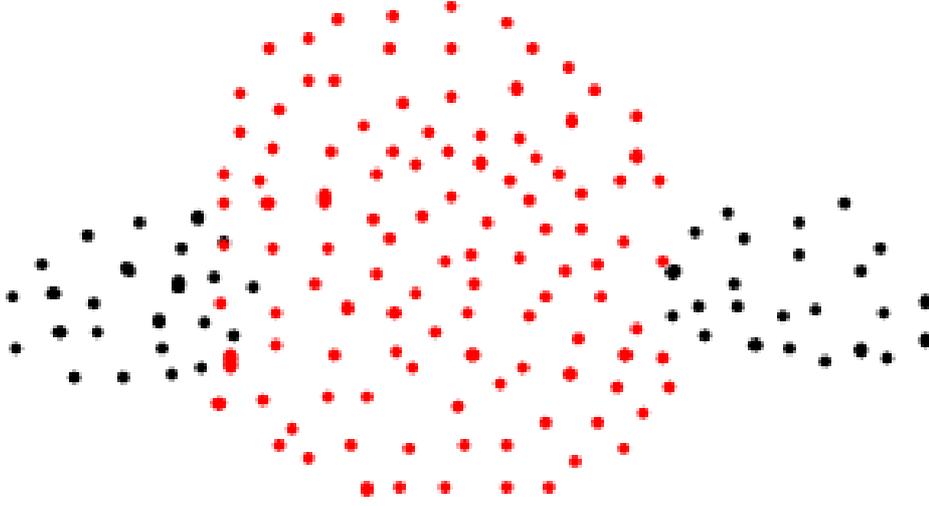
## 4.5.2 OLVQ-AEDP

A distância  $d_k$  (equação (4.6)) pode não descrever de forma precisa situações em que classes são formadas por intra-regiões. A Figura 4.2 apresenta duas classes: a classe vermelha é formada por uma região circular e a classe preta é formada por duas intra-regiões elípticas separadas pela classe vermelha.

Uma solução para isso é uma distância Euclidiana adaptativa  $d_m$  que seja diferente para cada protótipo  $m$ , pois os protótipos podem ser vistos como representantes das subregiões das classes. Para isso é necessário manter um vetor de pesos  $\boldsymbol{\lambda}_m$  para cada protótipo  $m$ . Esse vetor de pesos contém um peso para cada uma das variáveis e representa a influência das variáveis na dispersão dos dados representados pelo protótipo. A distância deve respeitar um critério  $J$  baseado na regra do vizinho mais próximo definido como:

$$J = \sum_{m=1}^M \sum_{i \in m} d_m(\mathbf{x}_i, \mathbf{w}_m(t)) \delta_{mi} \quad (4.11)$$

onde  $d_m$  é a distância Euclidiana adaptativa do protótipo  $m(t)$  entre esse protótipo e um padrão  $i$ , na iteração  $t$  tal que  $m$  é o protótipo mais próximo a  $i$  e  $\delta_{mi} = 1$  se  $i$  e  $m$  pertencem à mesma classe e  $\delta_{mi} = 0$ , caso contrário.



**Figura 4.2** Exemplo de classe formada por intra-regiões.

O critério  $J$ , apresentado na equação (4.11), representa a soma das distâncias entre os padrões  $i$  e seus protótipos vencedores  $m$  que pertencem à mesma classe. A distância Euclidiana adaptativa do protótipo  $m$  ( $d_m$ ) é calculada como segue:

$$d_m(\mathbf{x}_i, \mathbf{w}_m(t)) = \sum_{j=1}^p \lambda_m^j (x_i^j(t) - w_m^j(t))^2 \quad (4.12)$$

A adaptabilidade de  $d_m$  é expressa pelo vetor de pesos  $\boldsymbol{\lambda}_m = (\lambda_m^1, \dots, \lambda_m^p)$ , que representa a informação da dispersão da subregião de classe representada pelo protótipo  $m$ .

Os vetores de pesos  $\boldsymbol{\lambda}_m = (\lambda_m^1, \dots, \lambda_m^p)$  ( $m = 1, \dots, M$ ), que minimizam o critério da equação (4.11), seguindo as restrições  $\lambda_m^j > 0$  and  $\prod_{j=1}^p \lambda_m^j = 1$ , são atualizados conforme a seguinte expressão:

$$\lambda_m^j = \frac{\{\prod_{h=1}^p (\sum_{i \in m} (x_i^h - w_m^h)^2)\}^{\frac{1}{p}}}{\sum_{i \in m} (x_i^j - w_m^j)^2} \quad (4.13)$$

onde  $h = 1, \dots, p$  é o conjunto de índices das variáveis e  $i$  representa os padrões afetados por cada protótipo  $m$ . Então, para cada protótipo  $m$ , uma soma é calculada para cada variável  $j$

$$\Delta_m^j = \sum_{i \in m} (x_i^j - w_m^j)^2 \quad (4.14)$$

da diferença quadrática entre os valores do protótipo  $m$  e os padrões afetados por ele. Com isso, a equação (4.13) pode ser reescrita como:

$$\lambda_m^j = \frac{\{\prod_{h=1}^p (\Delta_m^h)\}^{\frac{1}{p}}}{\Delta_m^j} \quad (4.15)$$

A distância  $d_m$  deve capturar as subregiões internas de cada classe, representadas pelos protótipos. Se essas subregiões não forem sensíveis, essa distância deve se comportar de forma equivalente à distância Euclidiana padrão.

### O algoritmo

Assim como o classificador OLVQ-AEDC, o OLVQ-AEDP é um algoritmo online, então a soma  $\Delta_m^j$  definida 4.14 deve ser calculada como uma soma ponderada com a seguinte fórmula:

$$\Delta_m^j = ((1 - \alpha_m(t))\Delta_m^j(t-1)) + (\alpha_m(t)(x_i^j(t) - w_m^j(t))^2) \quad (4.16)$$

Os passos do algoritmo *OLVQ – AEDP* são os seguintes:

#### 1. Inicialização

- 1.1 Escolha  $M$  padrões distintos do conjunto de dados de treinamento  $\Omega$  aleatoriamente, gerando o subconjunto inicial de protótipos  $\{(\mathbf{w}_1(t), y_1), \dots, (\mathbf{w}_M(t), y_M)\}$  no instante  $t = 0$ .
- 1.2 De  $m = 1$  a  $M$  faça  $\alpha_m(t) = 0.3$  e para  $j = 1, \dots, p$  faça  $\lambda_m^j = 1$  e  $\Delta_m^j = 0$ .

#### 2. Escolha um padrão $i$ de $\Omega$ aleatoriamente com reposição

##### • Passo de atualização do protótipo:

- 2.1 Defina o protótipo vencedor  $c$  tal que

$$c = \arg \min_{m=1, \dots, M} d_m(\mathbf{x}_i, \mathbf{w}_m)$$

- 2.2 Se classe(i) = classe(c) faça  $\mathbf{w}_c(t+1) = \mathbf{w}_c(t) + \alpha_c(t)[\mathbf{x}_i - \mathbf{w}_c(t)]$

- 2.2.1 Atualize as somas  $\Delta_c^j$  para  $j = 1, \dots, p$ , usando a equação (4.16)

- 2.2.2 Para  $j = 1, \dots, p$ , compute  $\lambda_c^j$  com a equação (4.15)

2.3 Se classe(i)  $\neq$  classe(c) faça  $\mathbf{w}_c(t+1) = \mathbf{w}_c(t) - \alpha(t)[\mathbf{x}_i - \mathbf{w}_c(t)]$

2.4 Atualize

$$\alpha_c(t+1) = \frac{\alpha_c(t)}{1 + s(t)\alpha_c(t)}$$

onde  $s(t) = +1$  se  $i$  and  $c$  pertencem à mesma classe e  $s(t) = -1$ , caso contrário

3. **Critério de parada:** Se, para todo  $\alpha_m(t) \leq 0.00005$  ( $m = 1, \dots, M$ ), então PARE, senão vá para o passo 2.

# 5

## OLVQ Intervalar

*O verdadeiro conhecimento vem de dentro.*

—SÓCRATES

O algoritmo LVQ para dados intervalares introduzidos nessa dissertação é uma extensão do OLVQ de Kohonen (Kohonen, 2001), explicado no Capítulo 4. O fato de que o OLVQ é um algoritmo baseado em protótipos torna sua extensão de dados quantitativos para dados intervalares uma tarefa simples. Basta substituir a distância usada para achar o protótipo mais próximo por uma distância intervalar, e.g. uma distância Euclidiana intervalar.

Suponha que existam  $K$  classes rotuladas de  $1, \dots, K$ . Seja  $\mathfrak{S} = \{(\mathbf{x}_i, y_i)\}$  ( $i = 1, \dots, N$ ) um conjunto de dados simbólicos de treinamento. Cada padrão  $i$  é descrito por um vetor de  $p$  variáveis simbólicas  $\mathbf{x}_i = (x_i^1, \dots, x_i^p)$  e uma variável quantitativa discreta  $Y$  que toma valores no conjunto discreto  $G = \{1, \dots, K\}$ . Como visto no Capítulo 2, uma variável simbólica  $X^j$  ( $j = 1, \dots, p$ ) é uma variável intervalar quando, dado um padrão  $i$  de  $\mathfrak{S}$ ,  $X_i^j = [x_{iL}^j, x_{iU}^j] \subseteq \mathcal{A}_j$  onde  $\mathcal{A}_j = [L, U]$  é um intervalo.

Suponha  $M$  protótipos, tal que cada classe  $k$  ( $k = 1, \dots, K$ ) tem  $N_k$  protótipos com  $\sum_{k=1}^K N_k = M$ . Cada protótipo  $m$  da classe  $k$  é descrito por um vetor de  $p$  variáveis intervalares  $\mathbf{w}_{km} = (w_{km}^1, \dots, w_{km}^p)$ , onde  $w_{km}^j = [w_{kmL}^j, w_{kmU}^j]$  é um intervalo, e uma variável quantitativa discreta  $Y$  que toma valores no conjunto discreto  $G = \{1, \dots, K\}$ . Uma distância Euclidiana quadrática intervalar (de Carvalho *et al.*, 2006) entre o protótipo  $m$  da classe  $k$  e um padrão  $i$  é dada por

$$d(\mathbf{x}_i, \mathbf{w}_{km}) = \sum_{j=1}^p (x_{iL}^j - w_{kmL}^j)^2 + (x_{iU}^j - w_{kmU}^j)^2 \quad (5.1)$$

## 5.1. CASOS DA DISTÂNCIA PONDERADA GENERALIZADA PARA DADOS INTERVALARES

---

Aqui, o OLVQ aplicado a dados intervalares, usando a distância Euclidiana intervalar, é chamado de Interval Optimized Learning Vector Quantization (IOLVQ). Esse método, e outros que usam distância quadrática Euclidiana para achar o protótipo mais próximo, tendem a modelar melhor clusters e/ou classes de forma esférica. Isso ocorre porque pontos que são equidistantes a um protótipo são posicionados em um raio ao redor dele, segundo a distância Euclidiana.

Para permitir que o IOLVQ modele melhor conjuntos de dados com classes e sub-regiões de classes com diversas formas e tamanhos, uma distância quadrática Euclidiana ponderada e generalizada para dados intervalares pode ser adotada. Essa distância, que será mais detalhada adiante, é ponderada porque tem pesos que representam a influência das variáveis na distribuição dos dados, a influência dos protótipos nas suas classes e a importância de cada classe no conjunto de dados. Devido a esses pesos, a distância proposta tem 4 casos, que são definidos pela variação ou não dos pesos.

### 5.1 Casos da Distância Ponderada Generalizada para Dados Intervalares

A distância proposta tem três parâmetros: i) o peso  $\lambda$ , que representa o efeito da variável no protótipo; ii) o peso  $\gamma$ , que representa a influência do protótipo em sua classe; e, por último, iii) o peso  $\beta$  que representa a importância da classe no conjunto de dados. Isso gera quatro casos possíveis para a distância proposta.

1.  $\lambda = 1, \gamma = 1$  e  $\beta = 1$

Nesse caso, temos a distância Euclidiana intervalar tradicional. Esse caso tende a modelar melhor clusters e classes com formas esféricas.

2.  $\lambda > 0, \gamma = 1$  e  $\beta = 1$

Nesse caso, temos uma adaptação para dados intervalares da distância Euclidiana baseada em protótipo proposta por Silva Filho e Souza (2011). Esse caso modela classes com intraestruturas variadas e compostas de sub-regiões de diversas formas, mas com tamanhos similares.

3.  $\lambda > 0, \gamma > 0$  e  $\beta = 1$

Nesse caso, os pesos  $\lambda$  e  $\gamma$  podem variar, mas o peso  $\beta$  é forçado a ter o valor 1. Esse caso modela classes com intraestruturas variadas e compostas de sub-regiões de diversas formas e tamanhos. Isso é possível porque o peso  $\gamma$  representa a importância de cada protótipo em sua classe.

## 5.2. DISTÂNCIA QUADRÁTICA EUCLIDIANA PONDERADA GENERALIZADA PARA DADOS INTERVALARES

4.  $\lambda > 0$ ,  $\gamma > 0$  e  $\beta > 0$

Nesse caso, todos os parâmetros podem variar. Assim, esse caso é o mais geral da distância ponderada e o peso  $\beta$  representa a importância de cada classe no conjunto de dados. Esse caso modela classes de tamanhos desbalanceados com intraestruturas variadas e compostas de subregiões de diversas formas e tamanhos.

## 5.2 Distância Quadrática Euclidiana Ponderada Generalizada para Dados Intervalares

Considere que cada classe  $k$  ( $k = 1, \dots, K$ ) tem  $N_k$  protótipos. Sejam  $\boldsymbol{\lambda}_{km} = (\lambda_{km}^1, \dots, \lambda_{km}^p)$ ,  $\boldsymbol{\gamma}_k = (\gamma_{k1}, \dots, \gamma_{kN_k})$  e  $\boldsymbol{\beta} = (\beta_1, \dots, \beta_K)$  três vetores de pesos que satisfaçam as seguintes condições:

1.  $\lambda_{km}^j > 0$  e  $\prod_{j=1}^p \lambda_{km}^j = \gamma_{km}$  ( $m = 1, \dots, N_k$ );
2.  $\gamma_{km} > 0$  e  $\prod_{m=1}^{N_k} \gamma_{km} = \beta_k$ ;
3.  $\beta_k > 0$  e  $\prod_{k=1}^K \beta_k = 1$ .

Assim como no OLVQ-AEDP descrito no Capítulo 4, cada protótipo  $m$  da classe  $k$  tem uma distância diferente definida pelo vetor  $\boldsymbol{\lambda}_{km} = (\lambda_{km}^1, \dots, \lambda_{km}^p)$  e calculada no instante  $t$  como

$$d^G(\mathbf{x}_i, \mathbf{w}_{km}) = \sum_{j=1}^p \lambda_{km}^j [(x_{iL}^j - w_{kmL}^j)^2 + (x_{iU}^j - w_{kmU}^j)^2] \quad (5.2)$$

O vetor  $\boldsymbol{\lambda}_{km}$  modela a influência das variáveis na dispersão dos dados da subregião  $m$  da classe  $k$ . A partir do método dos Multiplicadores de Lagrange, os vetores de pesos  $\boldsymbol{\lambda}_{km} = (\lambda_{km}^1, \dots, \lambda_{km}^p)$  ( $m = 1, \dots, N_k$ ) são atualizados de acordo com a seguinte expressão:

$$\lambda_{km}^j = \frac{\{\gamma_{km} [\prod_{h=1}^p \Delta_{km}^h]\}^{\frac{1}{p}}}{\Delta_{km}^j} \quad (5.3)$$

onde  $\gamma_{km}$  modela o volume do protótipo  $km$  na sua classe e  $\Delta_{km}^j$  é a soma das diferenças quadráticas entre o protótipo  $km$  e os padrões corretamente afetados por ele. A soma  $\Delta_{km}^j$  é calculada como segue:

$$\Delta_{km}^j = \sum_{i \in km} [(x_{iL}^j - w_{kmL}^j)^2 + (x_{iU}^j - w_{kmU}^j)^2] \delta_{kmi} \quad (5.4)$$

onde  $\delta_{kmi} = 1$  se  $i$  pertence à mesma classe que  $km$  e  $\delta_{kmi} = 0$ , caso contrário.

Segundo o método dos Multiplicadores de Lagrange, o peso  $\gamma_{km}$  do protótipo  $m$  em sua classe  $k$  é atualizado de acordo com a seguinte expressão:

$$\gamma_{km} = \frac{\{\beta_k [\prod_{r=1}^{N_k} (\sum_{i \in kr} d^G(\mathbf{x}_i, \mathbf{w}_{kr}) \delta_{kri})]\}^{\frac{1}{N_k}}}{\sum_{i \in km} d^G(\mathbf{x}_i, \mathbf{w}_{km}) \delta_{kmi}} \quad (5.5)$$

onde  $\beta_k$  é o peso da classe  $k$  no conjunto de dados e  $\delta_{kmi} = 1$  se  $i$  pertence à mesma classe que  $km$  e  $\delta_{kmi} = 0$ , caso contrário.

A equação (5.5) pode resultar em erro caso  $\sum_{i \in km} d^G(\mathbf{x}_i, \mathbf{w}_{km}) \delta_{kmi} = 0$ , ou seja, se o protótipo não afetar padrão de treinamento algum. Para resolver esse problema, todos os protótipos que cumprem essa condição são removidos do conjunto de protótipos.

O peso  $\beta_k$  representa a importância da classe  $k$  em relação ao conjunto de dados. Ele é atualizado de acordo com a seguinte expressão:

$$\beta_k = \frac{\{\prod_{a=1}^K (\sum_{m \in a} \sum_{i \in am} d^G(\mathbf{x}_i, \mathbf{w}_{am}) \delta_{ami})\}^{\frac{1}{K}}}{\sum_{m \in k} \sum_{i \in km} d^G(\mathbf{x}_i, \mathbf{w}_{km}) \delta_{kmi}} \quad (5.6)$$

onde  $\delta_{kmi} = 1$  se  $i$  pertence à mesma classe que  $km$  e  $\delta_{kmi} = 0$ , caso contrário.

## 5.3 O Algoritmo

O algoritmo IOLVQ é composto por três grandes blocos de passos: um bloco de passos de treinamento online, um bloco batch de ajuste de pesos e um bloco de passos de validação. Dependendo de certas situações, o algoritmo pula de um bloco para o outro.

### 5.3.1 O bloco de passos de treinamento online

Durante esse bloco de passos, o algoritmo age como esperado para um algoritmo da família LVQ: ele seleciona instâncias aleatórias do conjunto de treinamento, define os protótipos vencedores e os ajusta, aproximando-os dos padrões afetados corretamente e os afastando dos padrões afetados incorretamente.

O IOLVQ também ajusta o vetor de pesos  $\lambda_{km}$  de cada protótipo vencedor  $km$  (exceto no caso 1 da distância proposta). Como esse bloco de passos é online, a soma  $\Delta_{km}^j$ , definida na equação (4.14), deve ser atualizada sempre que um protótipo  $km$  afetar um

padrão de treinamento  $i$  da mesma classe. Após isso, o vetor de pesos  $\boldsymbol{\lambda}_{km}$  deve ser recalculado.

Para resolver o problema da perda de informação da soma  $\Delta_{kc}^j$  (onde  $c$  é o índice do protótipo vencedor da classe  $k$ ), a atualização de  $\Delta_{kc}^j$  é uma soma ponderada da nova soma das diferenças quadráticas entre os limites inferiores e superiores do padrão de treinamento  $i$  e o seu protótipo vencedor  $kc(t)$  e a soma anterior  $\Delta_{kc}^j(t-1)$ , usando a taxa de aprendizagem  $\alpha_{kc}(t)$  do protótipo vencedor  $kc(t)$  como peso, resultando na equação (5.7).

$$\Delta_{kc}^j = [(1 - \alpha_{kc}(t))\Delta_{kc}^j(t-1)] + \{\alpha_{kc}(t)[(x_{iL}^j - w_{kcL}^j)^2 + (x_{iU}^j - w_{kcU}^j)^2]\} \quad (5.7)$$

O índice  $kc$  do protótipo mais próximo ao padrão  $i$  é achado usando a equação (5.8).

$$kc = \operatorname{argmin}\{d^G(\mathbf{x}_i, \mathbf{w}_{km}) \forall m \in (1, \dots, N_k) \forall k \in (k = 1, \dots, K)\} \quad (5.8)$$

Depois da atualização da soma  $\Delta_{kc}^j$ , o vetor de pesos  $\boldsymbol{\lambda}_{kc}$  deve ser atualizado, usando a equação (4.15).

Se o bloco de passos online for repetido um número de vezes igual a  $M \times n$ , onde  $M$  é o número de protótipos e  $n$  é o tamanho do conjunto de treinamento, o algoritmo pula para o próximo bloco de passos: o bloco batch de ajuste de pesos. Se todas as  $M$  taxas de aprendizagem dos protótipos se aproximarem de zero, o algoritmo para.

### 5.3.2 O bloco batch de ajuste de pesos

Se  $\beta = 1$  e  $\gamma = 1$  (casos 1 e 2), o algoritmo pula para o bloco de passos de validação. Senão, o algoritmo acha os protótipos vencedores  $kc$  de todas as instâncias do conjunto de dados de treinamento, atualizando para cada protótipo vencedor  $kc$  a soma  $\sum_{i \in kc} d^G(\mathbf{x}_i, \mathbf{w}_{kc}) \delta_{kci}$ , onde  $i$  é uma instância de treinamento afetada pelo protótipo  $kc$  e  $\delta_{kci} = 1$  se  $\text{classe}(i) = k$ .

Depois desse passo, se  $\beta \neq 1$ , os pesos  $\beta$  são atualizados de acordo com a equação (5.6) e se  $\gamma \neq 1$ , os pesos  $\gamma$  são atualizados usando a equação (5.5), nessa ordem. Essa ordem deve ser respeitada porque os pesos  $\beta$  são usados como restrições na fórmula dos pesos  $\gamma$ . Por último, o algoritmo pula para o bloco de passos de validação.

### 5.3.3 O bloco de passos de validação

Nesse bloco de passos, o algoritmo simplesmente passa por todo o conjunto de validação, achando os protótipos vencedores  $kc$  para cada instância de validação  $i$ . Se  $classe(i) \neq k$ , um erro de validação é computado. Se o número de erros de validação crescer três vezes seguidas, o algoritmo para. Senão, o algoritmo pula para o bloco de passos de treinamento online.

Deve-se destacar que o algoritmo sempre memoriza a última configuração dos seus protótipos. Dessa forma, quando o erro de validação crescer três vezes consecutivas, o algoritmo retornará o melhor conjunto de protótipos obtido antes de o erro crescer pela primeira das três vezes.

### 5.3.4 Passos do algoritmo

Os passos do IOLVQ estão listados abaixo.

#### 1. Inicialização

1.1 No instante  $t = 0$  De  $k = 1$  a  $K$ , escolha os  $N_k$  protótipos

$$\{(\mathbf{w}_{k1}(t), y_{k1}), \dots, (\mathbf{w}_{kN_k}(t), y_{kN_k})\}.$$

1.2 De  $k = 1$  a  $K$  De  $m = 1$  a  $N_k$ , faça  $\alpha_{km}(t) = 0.3$  e  $\gamma_{km} = 1$ .

1.3 De  $k = 1$  a  $K$  De  $m = 1$  a  $N_k$  e De  $j = 1$  a  $p$ , faça  $\lambda_{km}^j = 1$ .

1.4 De  $k = 1$  a  $K$ , faça  $\beta_k = 1$ .

#### 2. Bloco de passos de treinamento online

2.1 **Passo de classificação:** escolha aleatoriamente um padrão  $i$  do conjunto de treinamento  $\mathfrak{S}$ . Defina o protótipo vencedor  $kc$  usando a equação (5.8)

2.2 **Passo de atualização do protótipo e do vetor de pesos  $\lambda$ :**

2.2.1 Se  $classe(i) = k$ , faça

2.2.2.1 Atualize  $\Delta_{kc}^j, j \in (1, \dots, p)$  usando a equação (5.7).

2.2.2.2 Atualize o vetor de pesos  $\lambda_{kc}$  usando a equação (4.15).

2.2.2.3 De  $j = 1$  a  $p$ , faça  $w_{kc}^j(t+1) = w_{kc}^j(t) + \alpha_{kc}(t)[x_i^j - w_{kc}^j(t)]$ , onde a subtração dos intervalos  $x_i^j$  e  $w_{kc}^j(t)$  é calculada como  $x_i^j - w_{kc}^j(t) =$

$[x_{iL}^j - w_{kcL}^j, x_{iU}^j - w_{kcU}^j]$ , ou seja, é o intervalo formado pela subtração dos limites dos intervalos originais.

2.2.2 Se  $classe(i) \neq k$ , então, de  $j = 1$  a  $p$ , faça  $w_{kc}^j(t+1) = w_{kc}^j(t) - \alpha_{kc}(t)[x_i^j - w_{kc}^j(t)]$ .

2.3 **Passo de atualização da taxa de aprendizagem:** Compute  $\alpha_{kc}(t)$  de acordo com a seguinte equação:

$$\alpha_{kc}(t) = \frac{\alpha_{kc}(t-1)}{1 + s(t)\alpha_{kc}(t-1)} \quad (5.9)$$

onde  $s(t) = +1$  se  $i$  pertence à classe  $k$  e  $s(t) = -1$ , caso contrário.

2.4 **Crítério de parada:**

- Se todo  $\alpha_{km}(t) \leq 0.00005$  ( $m = 1, \dots, N_k$ ) ( $k = 1, \dots, K$ ), PARE.
- Se o bloco de passos 2 foi repetido  $M \times n$  vezes (onde  $M = \sum_{k=1}^K N_k$  e  $n$  é o tamanho do conjunto de treinamento), faça
  - \* Se  $\gamma_{km} \neq 1$  e/ou  $\beta_k \neq 1$  vá para o bloco 3
  - \* Caso contrário, vá para o bloco 4
- Senão, vá para o bloco 2

### 3. Bloco batch de ajuste de pesos

- 3.1 Para cada padrão  $i$  de  $\mathfrak{S}$ , defina o protótipo vencedor  $kc$  usando a equação (5.8) e atualize a soma  $\sum_{i \in kc} d^G(\mathbf{x}_i, \mathbf{w}_{kc}) \delta_{kci}$ .
- 3.2 Se  $\beta_k \neq 1$ , para cada classe ( $k = 1, \dots, K$ ) calcule seu respectivo peso no conjunto de treinamento  $\beta_k$  usando a equação (5.6).
- 3.3 Se  $\gamma_{km} \neq 1$ , para cada protótipo ( $m = 1, \dots, N_k$ ) ( $k = 1, \dots, K$ ) calcule seu respectivo peso  $\gamma_{km}$  na sua classe usando a equação (5.5).
- 3.4 Vá para o bloco 4

### 4. Bloco de passos de validação

- 4.1 Para cada instância  $i$  do conjunto de validação, defina o protótipo vencedor  $kc$  usando a equação (5.8). Se  $classe(i) \neq k$ , compute um erro de validação.

4.2 Se o número de erros de validação crescer três vezes consecutivas, PARE. Senão, vá para o bloco 2.

No Capítulo 6, o número de passos até a convergência é utilizado como uma das medidas de desempenho do IOLVQ com os diferentes casos da distância proposta. O número de passos representa quantas vezes o bloco de passos de treinamento online foi repetido até o algoritmo parar.

# 6

## Análise de Desempenho

*Todo os espíritos são invisíveis para os que não o possuem, e toda a avaliação é um produto do que é avaliado pela esfera cognitiva de quem avalia.*

—ARTHUR SCHOPENHAUER

Para comparar os desempenhos do algoritmo IOLVQ utilizando os quatro casos da distância, foram conduzidos experimentos com três conjuntos de dados intervalares sintéticos e um conjunto de dados reais de climas.

### 6.1 Resultados dos Conjuntos de Dados Sintéticos

As abordagens apresentadas dos diferentes casos da distância ponderada generalizada para intervalos são avaliadas de acordo com seus desempenhos de taxa de erros de classificação (em %) e de número de passos até a convergência do algoritmo. Essas medidas são computadas num experimento Monte Carlo com 100 replicações. Testes Anova e t de Student são aplicados aos valores obtidos de taxa de erro e testes de Wilcoxon para amostras independentes são aplicados aos valores obtidos de número de passos. A quantidade de protótipos designados para cada classe foi escolhido através de vários experimentos para determinar o melhor número.

Três conjuntos de dados sintéticos intervalares são considerados nessa dissertação: para cada caso da distância adaptativa proposta (exceto o primeiro caso) foi criado um conjunto sintético que demonstra as características daquele caso.

Para gerar esses conjuntos de dados, é preciso primeiro gerar conjuntos de dados quantitativos. Cada subregião das classes desses conjuntos clássicos foi desenhada de acordo com uma distribuição Gaussiana bivariada com componentes não-correlacionados.

Cada ponto  $(z_1, \dots, z_p)$  desses conjuntos de dados quantitativos sintéticos é utilizado para gerar um vetor de intervalos (um hipercubo de  $p$  dimensões, onde  $p$  é o número de variáveis e  $p = 2$  para todos os conjuntos dessa seção) definido como:

$$[(z_1 - \phi_1/2, z_1 + \phi_1/2), (z_2 - \phi_2/2, z_2 + \phi_2/2)].$$

Esses parâmetros  $\phi_1, \phi_2$  são selecionados aleatoriamente, para cada região de cada classe, do mesmo intervalo gerador predefinido. Todos as instâncias de dados de uma subregião de classe utilizam o mesmo par  $\phi_1, \phi_2$ . Isso é feito para preservar as formas das subregiões após a geração dos intervalos a partir dos pontos das distribuições Gaussianas. Os intervalos geradores usados foram:  $[1, 10], [1, 20], [1, 30]$  e  $[1, 40]$ .

### 6.1.1 Primeiro Conjunto de Dados Intervalares Sintéticos

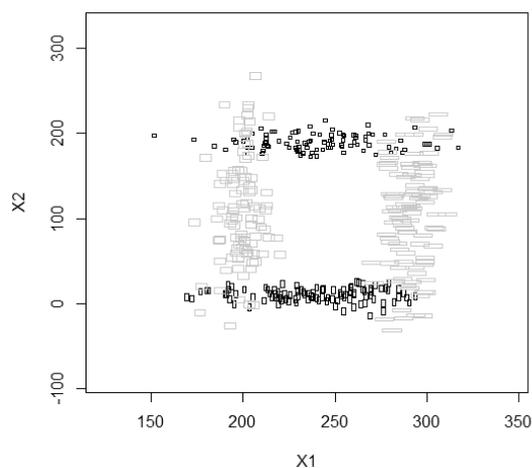
Este conjunto tem 400 retângulos distribuídos entre duas classes com estruturas diferentes e tamanhos idênticos (características modeladas pelo caso 2 da distância proposta  $d^G$ ): 200 retângulos para cada classe. Os parâmetros para esse conjunto são:

**Tabela 6.1** Parâmetros das distribuições Gaussianas

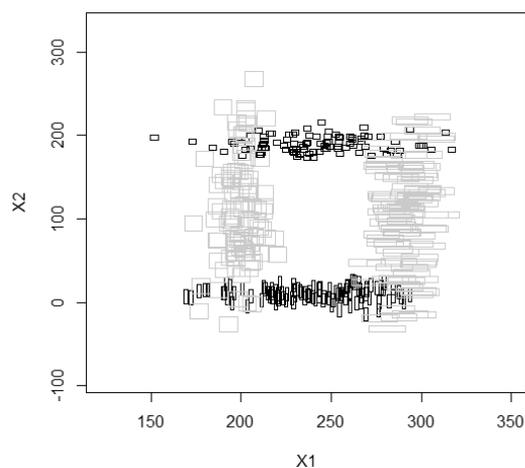
Classe 1				
$\mu_1 = 240$	$\mu_2 = 10$	$\sigma_1^2 = 900$	$\sigma_2^2 = 100$	$n = 100$
$\mu_1 = 240$	$\mu_2 = 190$	$\sigma_1^2 = 900$	$\sigma_2^2 = 100$	$n = 100$
Classe 2				
$\mu_1 = 200$	$\mu_2 = 105$	$\sigma_1^2 = 100$	$\sigma_2^2 = 3600$	$n = 100$
$\mu_1 = 290$	$\mu_2 = 105$	$\sigma_1^2 = 100$	$\sigma_2^2 = 3600$	$n = 100$

As figuras 6.1 a 6.4 mostram os conjuntos de dados intervalares gerados. As classes são representadas por grupos de retângulos de tons diferentes (preto para a classe 1 e cinza para a classe 2).

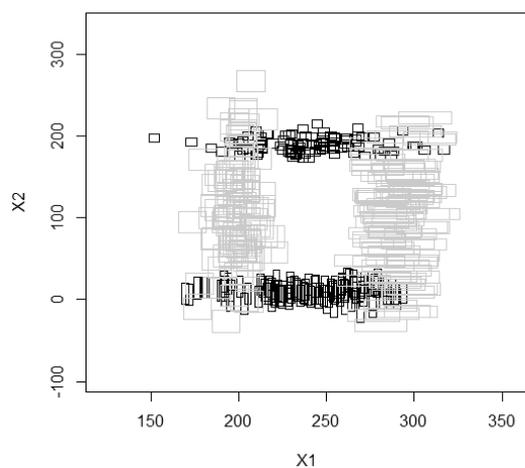
## 6.1. RESULTADOS DOS CONJUNTOS DE DADOS SINTÉTICOS



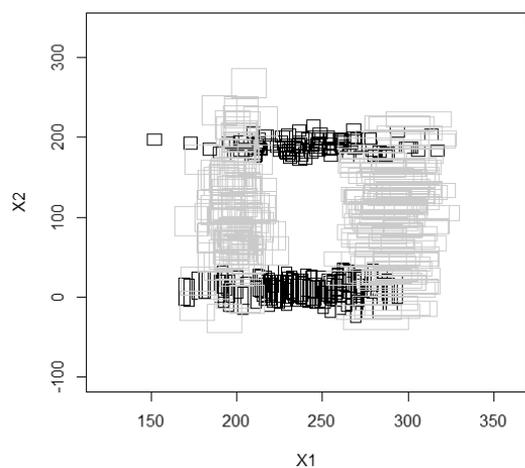
**Figura 6.1** Primeiro conjunto de dados sintéticos, considerando o intervalo [1,10].



**Figura 6.2** Primeiro conjunto de dados sintéticos, considerando o intervalo [1,20].



**Figura 6.3** Primeiro conjunto de dados sintéticos, considerando o intervalo [1,30].



**Figura 6.4** Primeiro conjunto de dados sintéticos, considerando o intervalo [1,40].

Os experimentos com esse conjunto utilizaram 5 protótipos para cada classe. A Tabela 6.2 contém os valores das médias e desvios padrões (entre parênteses) para as taxas de erros de classificação e os números de passos de treinamento. Os resultados nessa tabela demonstram os seguintes pontos:

1. O caso 2 da distância ponderada generalizada é superior ao caso 1 em todas as situações dos parâmetros  $\phi_1$  e  $\phi_2$ , em termos de taxa de erros de classificação, dado que cada classe do conjunto de dados tem duas subregiões com formatos elípticos e tamanhos iguais.

## 6.1. RESULTADOS DOS CONJUNTOS DE DADOS SINTÉTICOS

2. Como o caso 2 é uma situação particular dos casos 3 e 4, os resultados de taxa de erros dos casos 2, 3 e 4 foram equivalentes para todas as situações dos parâmetros  $\phi_1$  e  $\phi_2$ .
3. Quanto ao número de passos de treinamento, o caso 1 da distância proposta demonstra a menor média entre todos os casos.

**Tabela 6.2** Média e desvio padrão da taxa de erros de classificação, em %, e número de passos de treinamento para o primeiro conjunto de dados sintéticos

Distância ponderada generalizada		$\phi_1, \phi_2 \in$			
		[1,10]	[1,20]	[1,30]	[1,40]
1) $\lambda = 1, \gamma = 1, \beta = 1$	taxa de erros	16.6 (5.56)	16.17 (5.88)	15.3 (4.59)	15.46 (6.1)
	passos	5.21 (1.57)	5.38 (2.03)	5.07 (1.7)	4.93 (1.34)
2) $\lambda > 0, \gamma = 1, \beta = 1$	taxa de erros	11.94 (4.18)	11.69 (3.98)	10.29 (3.4)	9.33 (3.09)
	passos	5.11 (1.43)	5.54 (1.89)	5.57 (2.15)	5.37 (1.73)
3) $\lambda > 0, \gamma > 0, \beta = 1$	taxa de erros	11.17 (3.37)	10.93 (4.32)	10.41 (3.6)	8.95 (3.3)
	passos	5.71 (1.72)	5.51 (1.77)	5.55 (1.51)	5.21 (1.68)
4) $\lambda > 0, \gamma > 0, \beta > 0$	taxa de erros	11.21 (5.12)	11.55 (5.05)	10.16 (3.68)	9.16 (3.43)
	passos	5.63 (1.62)	5.89 (2.02)	5.67 (1.9)	5.45 (1.51)

A Tabela 6.3 mostra os valores das estatísticas dos testes t de Student e Anova, comparando os desempenhos do IOLVQ e os diferentes casos da distância com relação à taxa de erros de classificação. Esses valores demonstram que o caso 2 é melhor do que o caso 1 e equivalente aos casos 3 e 4, ao nível de significância de 5%. A Tabela 6.4 mostra os valores das estatísticas dos testes de Wilcoxon, comparando os desempenhos do IOLVQ e os diferentes casos da distância com relação ao número de passos de treinamento. O caso 1 se mostrou similar ao caso 2 e, na maioria das situações, superior aos casos 3 e 4, ao nível de significância de 5%.

## 6.1. RESULTADOS DOS CONJUNTOS DE DADOS SINTÉTICOS

**Tabela 6.3** Estatísticas dos testes de hipóteses para a taxa de erros de classificação do primeiro conjunto de dados sintéticos

Intervalo	Teste t de Student		Teste Anova	
	$H_0 : \mu_2 = \mu_1$ $H_1 : \mu_2 < \mu_1$	Decisão	$H_0 : \mu_2 = \mu_3 = \mu_4$ $H_1 : \mu_2 \neq \mu_3, \mu_2 \neq \mu_4 \text{ or } \mu_3 \neq \mu_4$	Decisão
[1,10]	-6.70	Rejeita $H_0$	1.02	Não rejeita $H_0$
[1,20]	-6.31	Rejeita $H_0$	0.82	Não rejeita $H_0$
[1,30]	-8.77	Rejeita $H_0$	0.12	Não rejeita $H_0$
[1,40]	-8.96	Rejeita $H_0$	0.34	Não rejeita $H_0$

**Tabela 6.4** Estatísticas dos testes de hipóteses para o número de passos do primeiro conjunto de dados sintéticos

Intervalo	$H_0 : \mu_1 = \mu_2$ $H_1 : \mu_1 \neq \mu_2$	Decisão	$H_0 : \mu_1 = \mu_3$ $H_1 : \mu_3 > \mu_1$	Decisão	$H_0 : \mu_1 = \mu_4$ $H_1 : \mu_4 > \mu_1$	Decisão
	[1,10]		4876		Não rejeita $H_0$	
[1,20]	5462	Não rejeita $H_0$	5536	Não rejeita $H_0$	6067	Rejeita $H_0$
[1,30]	5592	Não rejeita $H_0$	6301	Rejeita $H_0$	6117	Rejeita $H_0$
[1,40]	5641	Não rejeita $H_0$	5492	Não rejeita $H_0$	6184	Rejeita $H_0$

### 6.1.2 Segundo Conjunto de Dados Intervalares Sintéticos

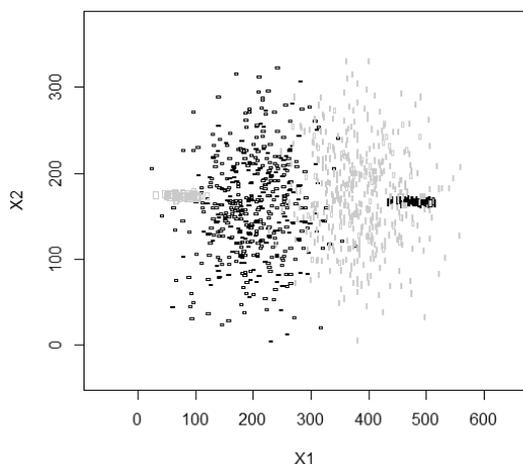
Este conjunto de dados possui 1000 retângulos distribuídos entre duas classes com estruturas e tamanhos iguais: 500 retângulos para cada classe. Cada classe é composta de uma pequena subregião elíptica com 50 retângulos e uma subregião esférica com 450 retângulos que sobrepõe a subregião elíptica da outra classe. Esse conjunto apresenta características que são modeladas pelo caso 3 da distância proposta  $d^G$ , pois tem classes de tamanhos idênticos e compostas de subregiões de formas e tamanhos diferentes. Os parâmetros para esse conjunto são:

**Tabela 6.5** Parâmetros das distribuições Gaussianas

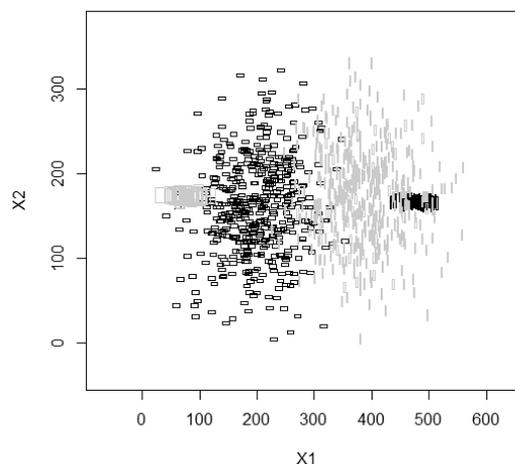
Classe 1				
$\mu_1 = 480$	$\mu_2 = 167$	$\sigma_1^2 = 400$	$\sigma_2^2 = 4$	$n = 50$
$\mu_1 = 200$	$\mu_2 = 160$	$\sigma_1^2 = 3600$	$\sigma_2^2 = 3600$	$n = 450$
Classe 2				
$\mu_1 = 80$	$\mu_2 = 173$	$\sigma_1^2 = 400$	$\sigma_2^2 = 4$	$n = 50$
$\mu_1 = 380$	$\mu_2 = 173$	$\sigma_1^2 = 3600$	$\sigma_2^2 = 3600$	$n = 450$

## 6.1. RESULTADOS DOS CONJUNTOS DE DADOS SINTÉTICOS

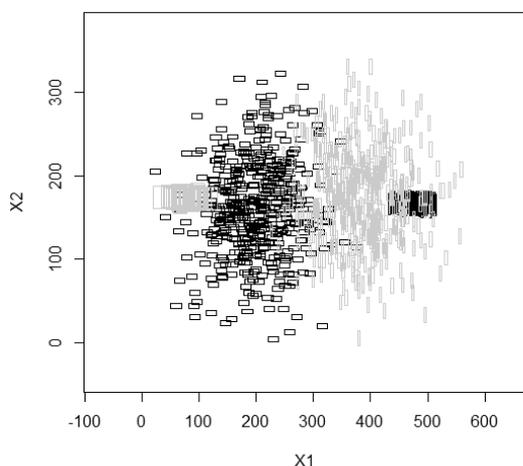
As figuras 6.5 a 6.8 mostram os conjuntos de dados intervalares gerados. As classes são representadas por grupos de retângulos de tons diferentes (preto para a classe 1 e cinza para a classe 2).



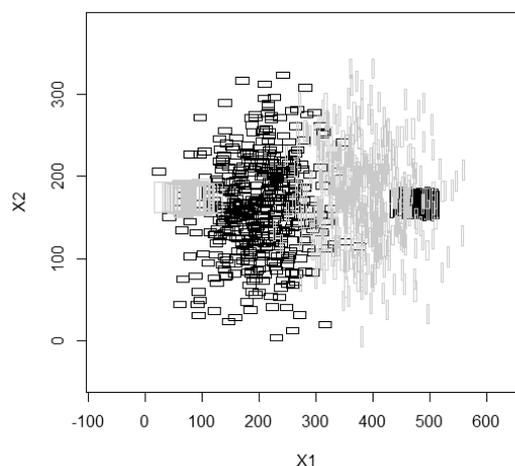
**Figura 6.5** Segundo conjunto de dados sintéticos, considerando o intervalo  $[1,10]$ .



**Figura 6.6** Segundo conjunto de dados sintéticos, considerando o intervalo  $[1,20]$ .



**Figura 6.7** Segundo conjunto de dados sintéticos, considerando o intervalo  $[1,30]$ .



**Figura 6.8** Segundo conjunto de dados sintéticos, considerando o intervalo  $[1,40]$ .

Os experimentos com esse conjunto utilizam 14 protótipos para cada classe. A Tabela 6.6 contém os valores das médias e desvios padrões (entre parênteses) para as taxas de erros de classificação e os números de passos de treinamento. Os resultados nessa tabela demonstram os seguintes pontos:

## 6.1. RESULTADOS DOS CONJUNTOS DE DADOS SINTÉTICOS

1. O caso 3 da distância ponderada generalizada é melhor do que os casos 1 e 2 para todas as situações dos parâmetros  $\phi_1$  e  $\phi_2$  em termos de taxa de erros. Isso ocorre porque cada classe do conjunto de dados tem duas subregiões com formas elípticas e tamanhos desbalanceados.
2. Como o caso 3 é uma situação particular do caso 4, os resultados de taxa de erros dos dois casos são similares para todas as situações dos parâmetros  $\phi_1$  e  $\phi_2$ .
3. Com relação ao número de passos de treinamento, os casos 2, 3 e 4 são apenas ligeiramente diferentes e o caso 1 é o que apresenta os melhores resultados.

**Tabela 6.6** Média e desvio padrão da taxa de erros de classificação, em %, e número de passos de treinamento para o segundo conjunto de dados sintéticos

Distância ponderada generalizada		$\phi_1, \phi_2 \in$			
		[1,10]	[1,20]	[1,30]	[1,40]
1) $\lambda = 1, \gamma = 1, \beta = 1$	taxa de erros	14.46 (3.6)	13.26 (3.26)	12.85 (2.78)	13.0 (3.03)
	passos	5.01 (1.49)	5.33 (1.51)	5.21 (1.87)	5.10 (1.67)
2) $\lambda > 0, \gamma = 1, \beta = 1$	taxa de erros	13.85 (3.57)	12.41 (2.89)	11.62 (3.5)	11.44 (3.56)
	passos	5.25 (1.81)	5.44 (1.77)	5.19 (1.6)	5.44 (1.93)
3) $\lambda > 0, \gamma > 0, \beta = 1$	taxa de erros	11.81 (3.25)	11.31 (3.18)	10.08 (2.92)	9.32 (2.88)
	passos	5.59 (1.4)	5.62 (1.33)	6.02 (1.49)	5.46 (1.42)
4) $\lambda > 0, \gamma > 0, \beta > 0$	taxa de erros	12.35 (3.38)	11.03 (3.63)	10.2 (3.03)	9.78 (3.34)
	passos	5.88 (1.38)	5.97 (1.47)	5.76 (1.37)	5.51 (1.37)

A Tabela 6.7 mostra os valores das estatísticas dos testes t de Student, comparando os desempenhos do IOLVQ e os diferentes casos da distância em relação à taxa de erros de classificação. Esses testes mostram que o caso 3 é superior ao caso 2 e o caso 2 é superior ao caso 1, ao nível de significância de 5%. Além disso, o caso 3 é tão bom quanto o caso 4. Em relação ao número de passos, os casos 1 e 2 são equivalentes e melhores que os casos 3 e 4, de acordo com as estatísticas dos testes de Wilcoxon, presentes na Tabela 6.8, ao nível de significância de 5%.

## 6.1. RESULTADOS DOS CONJUNTOS DE DADOS SINTÉTICOS

**Tabela 6.7** Estatísticas dos testes de hipóteses para a taxa de erros de classificação do segundo conjunto de dados sintéticos

Intervalo	$H_0 : \mu_3 = \mu_2$ $H_1 : \mu_3 < \mu_2$	Decisão	$H_0 : \mu_2 = \mu_1$ $H_1 : \mu_2 < \mu_1$	Decisão	$H_0 : \mu_3 = \mu_4$ $H_1 : \mu_3 \neq \mu_4$	Decisão
[1,10]	-4.23	Rejeita $H_0$	-1.20	Não rejeita $H_0$	-1.16	Não rejeita $H_0$
[1,20]	-2.57	Rejeita $H_0$	-1.94	Não rejeita $H_0$	0.58	Não rejeita $H_0$
[1,30]	-3.38	Rejeita $H_0$	-2.75	Não rejeita $H_0$	-0.30	Não rejeita $H_0$
[1,40]	-4.63	Rejeita $H_0$	-3.34	Não rejeita $H_0$	-1.04	Não rejeita $H_0$

**Tabela 6.8** Estatísticas dos testes de hipóteses para o número de passos do segundo conjunto de dados sintéticos

Intervalo	$H_0 : \mu_1 = \mu_2$ $H_1 : \mu_1 \neq \mu_2$	Decisão	$H_0 : \mu_1 = \mu_3$ $H_1 : \mu_3 > \mu_1$	Decisão	$H_0 : \mu_1 = \mu_4$ $H_1 : \mu_4 > \mu_1$	Decisão
[1,10]	5191	Não rejeita $H_0$	6567	Rejeita $H_0$	7187	Rejeita $H_0$
[1,20]	5043	Não rejeita $H_0$	5852	Rejeita $H_0$	6381	Rejeita $H_0$
[1,30]	5284	Não rejeita $H_0$	7042	Rejeita $H_0$	6686	Rejeita $H_0$
[1,40]	5547	Não rejeita $H_0$	6170	Rejeita $H_0$	6248	Rejeita $H_0$

### 6.1.3 Terceiro Conjunto de Dados Intervalares Sintéticos

O conjunto de dados tem 1000 retângulos distribuídos entre duas classes bastante desbalanceadas com diferentes estruturas: 40 retângulos para a classe 1 e 960 retângulos para a classe 2. A classe 1 é composta por uma subregião esférica com 10 retângulos e uma subregião elíptica com 30 retângulos. A classe 2 é composta de uma subregião esférica com 960 retângulos que sobrepõe completamente ambas as subregiões da classe 1. Esse conjunto apresenta características modeladas pelo caso 3 da distância proposta  $d^G$ , pois apresenta classes de tamanhos diferentes e compostas por subregiões de formas e tamanhos diferentes. Os parâmetros para esse conjunto são:

**Tabela 6.9** Parâmetros das distribuições Gaussianas

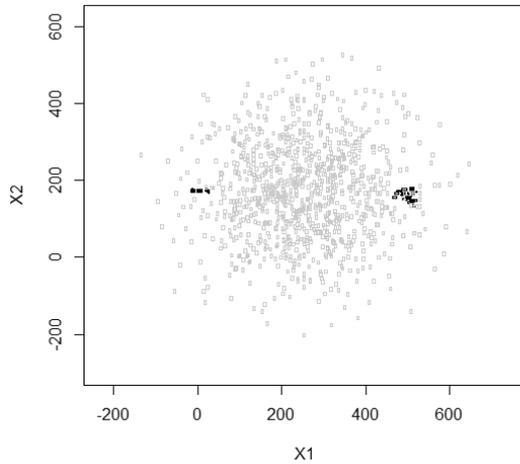
Classe 1				
$\mu_1 = 0$	$\mu_2 = 173$	$\sigma_1^2 = 225$	$\sigma_2^2 = 4$	$n = 10$
$\mu_1 = 500$	$\mu_2 = 160$	$\sigma_1^2 = 169$	$\sigma_2^2 = 169$	$n = 30$
Classe 2				
$\mu_1 = 253$	$\mu_2 = 167$	$\sigma_1^2 = 17424$	$\sigma_2^2 = 17424$	$n = 960$

As figuras 6.9 a 6.12 mostram os conjuntos de dados intervalares gerados. As classes

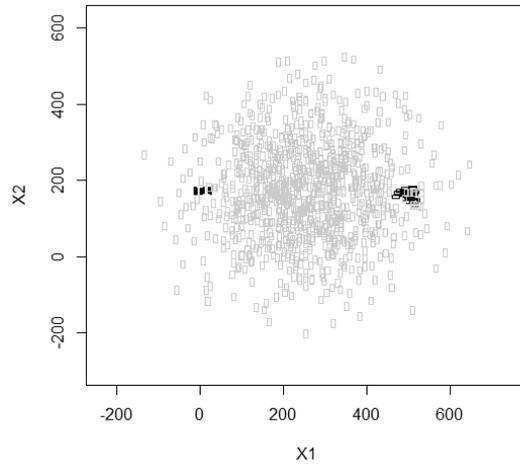
---

## 6.1. RESULTADOS DOS CONJUNTOS DE DADOS SINTÉTICOS

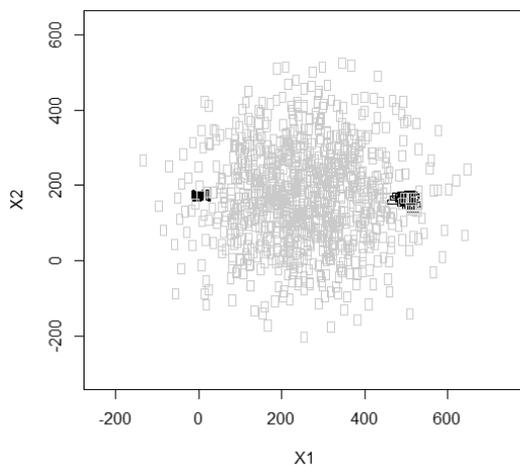
são representadas por grupos de retângulos de tons diferentes (preto para a classe 1 e cinza para a classe 2).



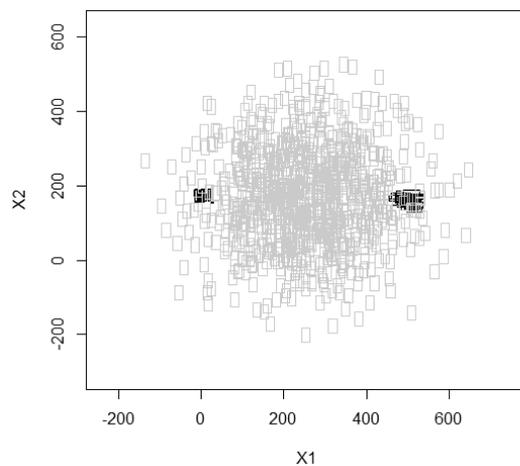
**Figura 6.9** Terceiro conjunto de dados sintéticos, considerando o intervalo [1,10].



**Figura 6.10** Terceiro conjunto de dados sintéticos, considerando o intervalo [1,20].



**Figura 6.11** Terceiro conjunto de dados sintéticos, considerando o intervalo [1,30].



**Figura 6.12** Terceiro conjunto de dados sintéticos, considerando o intervalo [1,40].

Os experimentos com esse conjunto utilizam 10 protótipos para cada classe. A Tabela 6.10 contém os valores das médias e desvios padrões (entre parênteses) para as taxas de erros de classificação e os números de passos de treinamento. Os resultados nessa tabela demonstram os seguintes pontos:

## 6.1. RESULTADOS DOS CONJUNTOS DE DADOS SINTÉTICOS

**Tabela 6.10** Média e desvio padrão da taxa de erros de classificação, em %, e número de passos de treinamento para o terceiro conjunto de dados sintéticos

Distância ponderada generalizada		$\phi_1, \phi_2 \in$			
		[1,10]	[1,20]	[1,30]	[1,40]
1) $\lambda = 1, \gamma = 1, \beta = 1$	taxa de erros	13.54 (4.0)	12.32 (4.05)	12.32 (3.68)	12.09 (4.12)
	passos	4.46 (1.05)	4.32 (0.68)	4.41 (0.85)	4.38 (0.69)
2) $\lambda > 0, \gamma = 1, \beta = 1$	taxa de erros	12.18 (3.84)	12.42 (3.65)	11.77 (3.57)	11.05 (3.05)
	passos	4.59 (0.99)	4.48 (0.78)	4.31 (0.61)	4.46 (0.95)
3) $\lambda > 0, \gamma > 0, \beta = 1$	taxa de erros	9.35 (2.78)	9.13 (3.01)	9.20 (3.11)	8.70 (2.56)
	passos	3.1 (0.58)	3.02 (0.65)	3.03 (0.59)	2.97 (0.54)
4) $\lambda > 0, \gamma > 0, \beta > 0$	taxa de erros	8.65 (2.95)	8.52 (2.86)	8.34 (2.68)	8.86 (3.1)
	passos	3.03 (0.54)	2.96 (0.53)	3.02 (0.53)	3.04 (0.58)

1. A média da taxa de erros do caso 4 é menor do que as dos casos 1 e 2, para todas as situações dos parâmetros  $\phi_1$  e  $\phi_2$ , e ligeiramente menor que a do caso 3.
2. Como esperado, os resultados de taxa de erros do caso 3 são melhores do que os dos casos 1 e 2, para todas as situações dos parâmetros  $\phi_1$  e  $\phi_2$ . Além disso os resultados de taxa de erros do caso 2 são melhores do que os do caso 1.
3. Quanto ao número de passos, os casos 3 e 4 tiveram performances melhores do que os casos 1 e 2. Isso se deve à maior complexidade desse conjunto de dados, o que faz com que o IOLVQ com os casos 1 e 2 da distância proposta demore mais para convergir.

A Tabela 6.11 mostra os valores das estatísticas dos testes t de Student, comparando os desempenhos do IOLVQ e os diferentes casos da distância com relação à taxa de erros de classificação. Os valores nessa tabela permitem concluir que os casos 3 e 4 tem médias de taxa de erros estatisticamente similares e melhores do que as dos casos 1 e 2, ao nível de significância de 5%. Para o número de passos, esse comportamento se repete, como se pode notar pelas estatísticas dos testes de Wilcoxon presentes na Tabela 6.12, também ao nível de significância de 5%. Além disso, os casos 1 e 2 tem desempenhos de número de passos similares.

## 6.2. RESULTADOS DO CONJUNTO DE DADOS DE CLIMA

**Tabela 6.11** Estatísticas dos testes de hipóteses para a taxa de erros de classificação do terceiro conjunto de dados sintéticos

Intervalo	$H_0 : \mu_4 = \mu_1$ $H_1 : \mu_4 < \mu_1$	Decisão	$H_0 : \mu_4 = \mu_2$ $H_1 : \mu_4 < \mu_2$	Decisão	$H_0 : \mu_4 = \mu_3$ $H_1 : \mu_4 \neq \mu_3$	Decisão
[1,10]	-9.84	Rejeita $H_0$	-7.30	Rejeita $H_0$	-1.72	Não rejeita $H_0$
[1,20]	-7.67	Rejeita $H_0$	-8.41	Rejeita $H_0$	-1.47	Não rejeita $H_0$
[1,30]	-8.73	Rejeita $H_0$	-7.67	Rejeita $H_0$	-2.09	Rejeita $H_0$
[1,40]	-6.27	Rejeita $H_0$	-5.04	Rejeita $H_0$	0.38	Não rejeita $H_0$

**Tabela 6.12** Estatísticas dos testes de hipóteses para o número de passos do terceiro conjunto de dados sintéticos

Intervalo	$H_0 : \mu_3 = \mu_4$ $H_1 : \mu_3 \neq \mu_4$	Decisão	$H_0 : \mu_3 = \mu_2$ $H_1 : \mu_3 < \mu_2$	Decisão	$H_0 : \mu_2 = \mu_1$ $H_1 : \mu_2 \neq \mu_1$	Decisão
[1,10]	4734	Não rejeita $H_0$	9317	Rejeita $H_0$	5519	Não rejeita $H_0$
[1,20]	4798	Não rejeita $H_0$	9338	Rejeita $H_0$	5583	Não rejeita $H_0$
[1,30]	4993	Não rejeita $H_0$	9336	Rejeita $H_0$	4957	Não rejeita $H_0$
[1,40]	5260	Não rejeita $H_0$	9554	Rejeita $H_0$	5015	Não rejeita $H_0$

## 6.2 Resultados do Conjunto de Dados de Clima

O tipo de variável simbólica que é o foco desse trabalho é a variável simbólica intervalar. Assim, um conjunto de dados simbólicos de climas foi extraído de uma página global da internet (cli, 2013) que contém observações meteorológicas oficiais, previsões do tempo e informações climatológicas de cidades selecionadas. Essas informações são fornecidas por serviços meteorológicos e hidrológicos nacionais ao redor do mundo.

Esse conjunto tem dados de 1415 cidades ou regiões de todos os continentes (exceto Antártida), com 17 variáveis, sendo 16 variáveis simbólicas intervalares e 1 variável quantitativa clássica, que representa a classificação climática de Köppen para aquela cidade (kcl, 2013). As classes são: Continental úmido, Deserto, Equatorial, Mediterrâneo, Monção, Oceânico, Savana, Semiárido, Subártico e Subtropical úmido.

As variáveis intervalares representam a variação de temperatura em cada mês (temperaturas mínima e máxima em °C) e a variação de precipitação em cada estação do ano (precipitações mínima e máxima em mm). Ao contrário dos dados presentes na Tabela 2.4, os intervalos presentes nesse conjunto de dados surgem naturalmente e não da agregação de uma base de dados clássica anterior. Para evitar problemas com a inversão de estações entre os hemisférios norte e sul, o primeiro mês de verão é considerado o primeiro mês do ano.

## 6.2. RESULTADOS DO CONJUNTO DE DADOS DE CLIMA

Todas as variáveis foram normalizadas entre 0 e 1. Uma amostra desse conjunto de dados é apresentada na Tabela 6.15 e na Tabela 6.16. Variáveis que representam a variação de temperatura mensal são denotadas de “temp”, seguido do índice numérico do mês. Variáveis que representam variação de precipitação sazonal são denotadas de “prec”, seguido do nome da estação.

Os casos da distância proposta para o IOLVQ tiveram seus desempenhos de taxa de erros de classificação e número de passos de treinamento avaliados com o conjunto de dados de clima. Essa avaliação foi efetuada utilizando um experimento Monte Carlo de 10 repetições com uma validação cruzada com 10 partições para cada repetição, gerando um total de 100 valores de taxas de erros e números de passos.

O número de protótipos escolhidos para cada classe equivale a cerca de 1/10 do tamanho da respectiva classe. Assim, foram designados 17 protótipos para o clima continental úmido, 14 protótipos para o clima de deserto, 6 protótipos para o clima equatorial, 10 protótipos para o clima mediterrâneo, 6 protótipos para o clima de monção, 21 protótipos para o clima oceânico, 19 protótipos para o clima de savana, 17 protótipos para o clima semiárido, 3 protótipos para o clima subártico e 22 protótipos para o clima subtropical úmido.

A Tabela 6.13 mostra os valores médios e os desvios padrões para taxa de erros de classificação, em %, e número de passos para os quatro casos da distância ponderada generalizada. Os resultados nessa tabela demonstram os seguintes pontos:

1. O caso 2 da distância ponderada generalizada é melhor do que os casos 1, 3 e 4, em termos de taxa de erros.
2. Com relação ao número de passos de treinamento, o caso 3 da distância ponderada generalizada é melhor do que os casos 1, 2 e 4.

**Tabela 6.13** Média e desvio padrão da taxa de erros de classificação, em %, e número de passos de treinamento para o conjunto de dados de clima

Métrica	1) $\lambda = 1, \gamma = 1, \beta = 1$	2) $\lambda > 0, \gamma = 1, \beta = 1$	3) $\lambda > 0, \gamma > 0, \beta = 1$	4) $\lambda > 0, \gamma > 0, \beta > 0$
taxa de erros	24.68 (3.34)	23.47 (3.40)	24.05 (3.46)	24 (2.90)
passos	4.99 (1.22)	5.03 (1.37)	4.86 (1.17)	4.95 (1.12)

De acordo com os valores da Tabela 6.14, de fato o caso 2 tem um desempenho melhor do que o do caso 1 e tão bom quanto os dos casos 3 e 4, em termos de média

## 6.2. RESULTADOS DO CONJUNTO DE DADOS DE CLIMA

---

de taxa de erros de classificação, a um nível de significância de 5%. Quanto ao número médio de passos, todos os casos são similares, segundo a estatística do teste de Kruskal-Wallis. Para finalizar, a aplicação do algoritmo com a distância proposta ao conjunto de dados intervalares de clima demonstra a importância de utilizar uma distância localmente ponderada, para medir a distância entre padrões e protótipos.

**Tabela 6.14** Estatísticas dos testes t de Student e Kruskal-Wallis para as médias de taxa de erros e número de passos, respectivamente.

Média da taxa de erros		Média do número de passos	
Hipóteses	Estatísticas/ Decisão	Hipóteses	Estatísticas/ Decisão
$H_0 : \mu_1 = \mu_2$ $H_1 : \mu_1 > \mu_2$	2.55 Rejeita $H_0$	$H_0 : \mu_1 = \mu_2 = \mu_3 = \mu_4$ $H_1 : \mu_1 \neq \mu_2, \mu_1 \neq \mu_3,$ $\mu_1 \neq \mu_4, \mu_2 \neq \mu_3,$ $\mu_2 \neq \mu_4$ ou $\mu_3 \neq \mu_4$	7.38 Não Rejeita $H_0$
$H_0 : \mu_3 = \mu_2$ $H_1 : \mu_3 > \mu_2$	1.20 Não Rejeita $H_0$		
$H_0 : \mu_4 = \mu_2$ $H_1 : \mu_4 > \mu_2$	1.20 Não Rejeita $H_0$		

---

## 6.2. RESULTADOS DO CONJUNTO DE DADOS DE CLIMA

---

**Tabela 6.15** Amostra do conjunto de dados climáticos.

u	Cidade	Clima
1	Calgary	Continental úmido
2	Charlottetown	Continental úmido
3	Cranbrook	Continental úmido
4	Edmonton	Continental úmido
5	Hermosillo	Deserto
6	Mexicali	Deserto
7	Saltillo	Deserto
8	Torreon	Deserto
9	Belém	Equatorial
10	Florencia	Equatorial
11	Ibague	Equatorial
12	Leticia	Equatorial
13	Algiers	Mediterrâneo
14	Annaba	Mediterrâneo
15	Constantine	Mediterrâneo
16	Setif	Mediterrâneo
17	Nassau	Monção
18	Medellin	Monção
19	Heredia	Monção
20	Quepos	Monção
21	Vancouver	Oceânico
22	Victoria	Oceânico
23	Bogotá	Oceânico
24	Manizales	Oceânico
25	Lagos	Savana
26	Brasília	Savana
27	Cuiabá	Savana
28	Fortaleza	Savana
29	Luanda	Semi árido
30	Tripoli	Semi árido
31	Dakar	Semi árido
32	Agra	Semi árido
33	Banff	Subártico
34	Inuvik	Subártico
35	Anchorage	Subártico
36	Dudinka	Subártico
37	Nova Iorque	Subtropical úmido
38	Durban	Subtropical úmido
39	Liverpool	Subtropical úmido
40	Florianópolis	Subtropical úmido

---

---

6.2. RESULTADOS DO CONJUNTO DE DADOS DE CLIMA

---

**Tabela 6.16** Amostra do conjunto de dados climáticos.

$w_u$	temp-1	...	temp-12	prec-summer	...	prec-spring
$w_1$	[0.12,0.41]	...	[0.25,0.50]	[0.05,0.07]	...	[0.02,0.08]
$w_2$	[0.17,0.40]	...	[0.26,0.45]	[0.07,0.08]	...	[0.12,0.14]
$w_3$	[0.14,0.44]	...	[0.28,0.52]	[0.03,0.05]	...	[0.03,0.06]
$w_4$	[0.13,0.42]	...	[0.25,0.51]	[0.06,0.08]	...	[0.02,0.07]
$w_5$	[0.49,0.85]	...	[0.53,0.85]	[0.01,0.08]	...	[0.01,0.01]
$w_6$	[0.44,0.86]	...	[0.50,0.84]	[0.00,0.01]	...	[0.00,0.01]
$w_7$	[0.32,0.63]	...	[0.47,0.74]	[0.03,0.06]	...	[0.01,0.05]
$w_8$	[0.43,0.76]	...	[0.54,0.85]	[0.03,0.03]	...	[0.00,0.03]
$w_9$	[0.45,0.46]	...	[0.79,0.79]	[0.19,0.36]	...	[0.16,0.20]
$w_{10}$	[0.45,0.68]	...	[0.60,0.77]	[0.09,0.16]	...	[0.33,0.43]
$w_{11}$	[0.38,0.60]	...	[0.54,0.70]	[0.06,0.10]	...	[0.19,0.33]
$w_{12}$	[0.47,0.65]	...	[0.61,0.77]	[0.25,0.31]	...	[0.33,0.42]
$w_{13}$	[0.32,0.59]	...	[0.42,0.64]	[0.00,0.01]	...	[0.06,0.08]
$w_{14}$	[0.33,0.58]	...	[0.43,0.63]	[0.00,0.01]	...	[0.05,0.09]
$w_{15}$	[0.30,0.63]	...	[0.39,0.64]	[0.01,0.02]	...	[0.06,0.08]
$w_{16}$	[0.32,0.62]	...	[0.40,0.62]	[0.01,0.02]	...	[0.05,0.06]
$w_{17}$	[0.48,0.66]	...	[0.58,0.74]	[0.14,0.20]	...	[0.08,0.15]
$w_{18}$	[0.34,0.59]	...	[0.51,0.70]	[0.10,0.14]	...	[0.18,0.28]
$w_{19}$	[0.31,0.52]	...	[0.48,0.67]	[0.17,0.27]	...	[0.06,0.46]
$w_{20}$	[0.48,0.65]	...	[0.62,0.77]	[0.36,0.40]	...	[0.06,0.54]
$w_{21}$	[0.21,0.39]	...	[0.35,0.50]	[0.03,0.05]	...	[0.09,0.16]
$w_{22}$	[0.17,0.39]	...	[0.32,0.50]	[0.02,0.03]	...	[0.05,0.11]
$w_{23}$	[0.14,0.31]	...	[0.35,0.49]	[0.04,0.05]	...	[0.09,0.14]
$w_{24}$	[0.24,0.43]	...	[0.42,0.58]	[0.06,0.08]	...	[0.19,0.23]
$w_{25}$	[0.47,0.62]	...	[0.62,0.77]	[0.10,0.27]	...	[0.11,0.28]
$w_{26}$	[0.35,0.35]	...	[0.70,0.69]	[0.18,0.21]	...	[0.07,0.33]
$w_{27}$	[0.48,0.48]	...	[0.82,0.77]	[0.17,0.18]	...	[0.08,0.21]
$w_{28}$	[0.51,0.51]	...	[0.76,0.76]	[0.04,0.19]	...	[0.02,0.03]
$w_{29}$	[0.49,0.60]	...	[0.62,0.72]	[0.02,0.03]	...	[0.00,0.04]
$w_{30}$	[0.41,0.64]	...	[0.50,0.69]	[0.00,0.00]	...	[0.01,0.04]
$w_{31}$	[0.48,0.60]	...	[0.57,0.67]	[0.01,0.20]	...	[0.00,0.00]
$w_{32}$	[0.61,0.88]	...	[0.68,0.97]	[0.05,0.21]	...	[0.01,0.02]
$w_{33}$	[0.08,0.37]	...	[0.22,0.46]	[0.05,0.05]	...	[0.03,0.05]
$w_{34}$	[0.08,0.35]	...	[0.10,0.28]	[0.02,0.03]	...	[0.01,0.02]
$w_{35}$	[0.15,0.33]	...	[0.26,0.42]	[0.02,0.05]	...	[0.02,0.03]
$w_{36}$	[0.00,0.16]	...	[0.00,0.15]	[0.03,0.05]	...	[0.04,0.05]
$w_{37}$	[0.33,0.52]	...	[0.40,0.56]	[0.07,0.08]	...	[0.13,0.14]
$w_{38}$	[0.41,0.56]	...	[0.53,0.66]	[0.09,0.12]	...	[0.10,0.15]
$w_{39}$	[0.32,0.58]	...	[0.45,0.66]	[0.06,0.08]	...	[0.06,0.11]
$w_{40}$	[0.42,0.44]	...	[0.62,0.65]	[0.13,0.17]	...	[0.18,0.18]

---

# 7

## Conclusões

*Os bons trabalhadores têm sempre a ideia de que  
ainda poderiam trabalhar mais.*

—ANDRÉ GIDE

Esse trabalho introduziu uma extensão do OLVQ de Kohonen para dados intervalares. Os dados intervalares foram escolhidos para serem estudados devido a sua capacidade de modelar variabilidade e incerteza. O OLVQ foi escolhido para ser estendido para dados intervalares porque, devido a sua estrutura, essa tarefa se resume a substituir a medida utilizada pelo algoritmo para calcular a distância entre os padrões e os protótipos.

Uma distância Euclidiana intervalar ponderada e generalizada foi introduzida para ser utilizada na extensão do OLVQ de Kohonen para dados intervalares. A distância proposta tem quatro casos. O primeiro caso é a distância Euclidiana intervalar e tende a modelar clusters e classes com formas esféricas. O segundo caso é uma distância intervalar baseada em protótipos que leva em consideração a influência de cada variável na distribuição dos dados representados por cada protótipo. Assim, o segundo caso é capaz de modelar clusters e classes com diversas formas, mas tamanhos similares. O terceiro caso permite à distância lidar com subregiões com diversas formas e tamanhos dentro das classes. O último caso da distância é o mais geral de todos e é capaz de modelar classes desbalanceadas compostas de subregiões com formas e tamanhos diferentes. O algoritmo OLVQ que utiliza a distância proposta foi chamado de Interval Optimized Learning Vector Quantization (IOLVQ).

O algoritmo teve seu desempenho, em termos de taxa de erros de classificação e número de passos até a convergência, avaliado usando os quatro casos da distância proposta, aplicados a três conjuntos de dados intervalares sintéticos e um conjunto

real. Os experimentos confirmaram a utilidade de uma distância ponderada localmente, demonstrando melhores resultados de taxa de erros dos casos mais gerais da distância. Quanto ao número de passos de treinamento, os casos menos gerais tiveram melhor desempenho com conjuntos de dados mais simples, mas seus desempenhos tendem a se deteriorar com conjuntos complexos.

## 7.1 Trabalhos futuros

Apesar dos bons resultados obtidos, é possível notar alguns pontos de melhora no método. O primeiro deles é a inicialização dos protótipos. A inicialização dos protótipos é um problema compartilhado por todos os métodos baseados em protótipos, pois diferentes inicializações podem gerar resultados bastante diferentes. Portanto, é interessante buscar uma forma de otimizar a escolha dos protótipos, bem como o número de protótipos escolhidos, para obter resultados melhores.

Outro ponto é que os pesos da distância são calculados em duas etapas diferentes. Os pesos  $\lambda$  das variáveis são calculados online, durante o treinamento do algoritmo, e os pesos  $\gamma$  dos protótipos e  $\beta$  das classes são calculados de forma batch, separados do treinamento. Isso pode gerar perda de informação e a heurística utilizada para diminuir esse problema, apresentada na equação (5.7), se provou útil nos experimentos. Porém, o ideal seria buscar uma forma de calcular esses pesos de uma forma unificada, evitando perda de informação.

Também é interessante estudar a extensão para dados clássicos da abordagem proposta. Para isso, basta substituir a distância Euclidiana intervalar por uma distância clássica.

# Referências Bibliográficas

- (2013). Köppen Climate Classification - Encyclopædia Britannica. <http://goo.gl/G4Uep>.
- (2013). World Meteorological Organization. <http://goo.gl/r2Sfb>.
- Aly, M. (2005). Survey on Multi-Class Classification Methods.
- Appice, A., D'Amato, C., Esposito, F., e Malerba, D. (2006). Classification of Symbolic Objects: A Lazy Learning Approach. *Intelligent Data Analysis*, **10**(4), 301–324.
- Beheshti, M., Berrached, A., De Korvin, A., Hu, C., e Sirisaengtaksin, O. (1998). On Interval Weighted Three-layer Neural Networks. In *In Proceedings of the 31 Annual Simulation Symposium*, pages 188–194. IEEE Computer Society Press.
- Borgelt, C., Nürnberger, A., e Kruse, R. (2005). Fuzzy Learning Vector Quantization with Size and Shape Parameters. *The 14th IEEE International Conference on Fuzzy Systems (FUZZ)*, pages 195–200.
- de Carvalho, F. A. (2007). Fuzzy c-Means Clustering Methods for Symbolic Interval Data. *Pattern Recogn. Lett.*, **28**(4), 423–437.
- de Carvalho, F. A., Brito, P., e Bock, H. H. (2006). Dynamic Clustering for Interval Data Based on L2 Distance. *Computational Statistics*, **21**, 231–250.
- Diday, E. e Billard, L. (2006). *Symbolic Data Analysis: Conceptual Statistics and Data Mining*. Wiley Interscience.
- Diday, E. e Noirhomme-Fraiture, M. (2008). *Symbolic Data Analysis and the SODAS Software*. Wiley Interscience.
- D'Oliveira, S. T., de Carvalho, F. A., e Souza, R. M. (2004). Classification of SAR Images Through a Convex Hull Region Oriented Approach. In N. Pal, N. Kasabov, R. Mudi, S. Pal, e S. Parui, editors, *Neural Information Processing*, volume 3316 of *Lecture Notes in Computer Science*, pages 769–774. Springer Berlin / Heidelberg.
- Hart, P. (1968). The Condensed Nearest Neighbor Rule. *Information Theory, IEEE Transactions on*, **14**(3), 515 – 516.
- Hastie, T., Tibshirani, R., e Friedman, J. (2001). *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. Springer.

- Ichino, M., Yaguchi, H., e Diday, E. (1996). A Fuzzy Symbolic Pattern Classifier, In. *Ordinal and Symbolic Data Analysis*, pages 92–102.
- Ishibuchi, H., Tanaka, H., e Okada, H. (1993). An Architecture of Neural Networks with Interval Weights and its Application to Fuzzy Regression Analysis. *Fuzzy Sets and Systems*, **57**(1), 27 – 39.
- Karayiannis, N. B. e Randolph-Gips, M. M. (2003). Soft Learning Vector Quantization and Clustering Algorithms Based on Non-Euclidean Norms: Multinorm Algorithms. *IEEE Transactions on Neural Networks*, pages 89–102.
- Kietzmann, T. C., Lange, S., e Riedmiller, M. (2008). Incremental GRLVQ: Learning Relevant Features for 3D Object Recognition. *Neurocomputing*, pages 2868–2879.
- Kohonen, T. (2001). *Self-Organizing Maps*. Springer, third edition.
- Kubat, M. e Matwin, S. (1997). Addressing the Curse of Imbalanced Training Sets: One-Sided Selection. In *In Proceedings of the Fourteenth International Conference on Machine Learning*, pages 179–186. Morgan Kaufmann.
- Paredes, R. e Vidal, E. (2004). Learning Prototypes and Distances (LPD). A Prototype Reduction Technique Based on Nearest Neighbor Error Minimization. *International Conference on Pattern Recognition*, **3**, 442–445.
- Roque, A. M., Maté, C., Arroyo, J., e Sarabia, A. (2007). iMLP: Applying Multi-Layer Perceptrons to Interval-Valued Data. *Neural Processing Letters*, **25**(2), 157–169.
- Rossi, F. e Conan-guez, B. (2002). Multi-layer Perceptron on Interval Data, in. *Classification, Clustering and Data Analysis (IFCS, 2002)*, 427–434.
- Schneider, P., Biehl, M., e Hammer, B. (2009). Distance Learning in Discriminative Vector Quantization. *Neural Computation*, **21**(10), 2942–2969.
- Silva, A. e Brito, P. (2006). Linear Discriminant Analysis for Interval Data. *Computational Statistics*, **21**, 289–308.
- Silva Filho, T. M. e Souza, R. M. (2011). Pattern Classifiers with Adaptive Distances. *International Joint Conference on Neural Networks (IJCNN)*, pages 1508–1514.
- Souza, R. M., Queiroz, D. C., e Cysneiros, F. J. (2011). Logistic Regression-Based Pattern Classifiers for Symbolic Interval Data. *Pattern Anal. Appl.*, **14**(3), 273–282.

Tomek, I. (1976). Two Modifications of CNN. *Systems, Man and Cybernetics, IEEE Transactions on*, **SMC-6**(11), 769 –772.

Wilson, D. L. (1972). Asymptotic Properties of Nearest Neighbor Rules Using Edited Data. *Systems, Man and Cybernetics, IEEE Transactions on*, **SMC-2**(3), 408 –421.