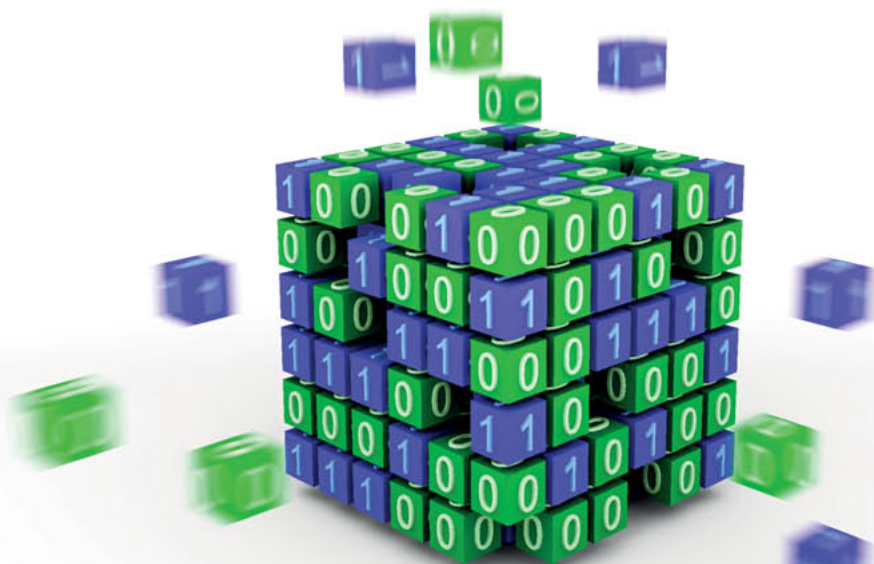


João Gama • André Ponce de Leon Carvalho
Katti Faceli • Ana Carolina Lorena • Márcia Oliveira

Extração de Conhecimento de Dados

Data Mining



2ª Edição
Revista e aumentada


EDIÇÕES SÍLABO

Extração de Conhecimento de Dados

Data Mining

JOÃO GAMA
ANDRÉ PONCE DE LEON CARVALHO
KATTI FACELI
ANA CAROLINA LORENA
MÁRCIA OLIVEIRA



É expressamente proibido reproduzir, no todo ou em parte, sob qualquer forma ou meio, **NOMEADAMENTE FOTOCÓPIA**, esta obra. As transgressões serão passíveis das penalizações previstas na legislação em vigor.

Visite a Sílabo na rede
www.silabo.pt

Este trabalho é financiado por Fundos FEDER através do Programa Operacional Fatores de Competitividade – COMPETE e por Fundos Nacionais através da FCT – Fundação para a Ciência e a Tecnologia no âmbito do projeto «FCOMP-01-0124-FEDER-022701».

This work is funded by the ERDF – European Regional Development Fund through the COMPETE Programme (operational programme for competitiveness) and by National Funds through the FCT – Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within project «FCOMP-01-0124-FEDER-022701»

Editor: Manuel Robalo

FICHA TÉCNICA:

Título: Extração de Conhecimento de Dados – *Data Mining*

Autores: João Gama, André Ponce de Leon Carvalho, Katti Faceli,
Ana Carolina Lorena, Márcia Oliveira

© Edições Sílabo, Lda.

Capa: Pedro Mota

1ª Edição – Lisboa, outubro de 2012.

2ª Edição – Lisboa, setembro de 2015.

Impressão e acabamentos: Cafilesa – Soluções Gráficas, Lda.

Depósito Legal: 396214/15

ISBN: 978-972-618-811-7

EDIÇÕES SÍLABO, LDA.

R. Cidade de Manchester, 2

1170-100 Lisboa

Tel.: 218130345

Fax: 218166719

e-mail: silabo@silabo.pt

www.silabo.pt

Índice

1	Introdução	9
1.1	Inteligência Artificial e Aprendizagem Automática	10
1.2	Indução de Hipóteses	12
1.3	Viés Indutivo	13
1.4	Tarefas de aprendizagem	14
1.5	Estrutura do Livro	15
I	Preparação de Dados	17
2	Análise Exploratória de Dados	21
2.1	Caraterização de Dados	21
2.2	Exploração de Dados	26
2.3	Considerações Finais	38
3	Pré-processamento de Dados	41
3.1	Teoria da Informação	42
3.2	Eliminação Manual de Atributos	43
3.3	Integração de Dados	44
3.4	Amostragem de Dados	44
3.5	Dados Desbalanceados	46
3.6	Limpeza de Dados	47
3.7	Transformação de Dados	55
3.8	Redução de Dimensionalidade	60
3.9	Considerações Finais	66
II	Modelos Preditivos	67
4	Métodos Baseados em Distâncias	73
4.1	Introdução	73

4.2	O Algoritmo do 1-Vizinho Mais Próximo	74
4.3	O Algoritmo k -NN	77
4.4	Discussão: Vantagens e Desvantagens	78
4.5	Desenvolvimentos	80
4.6	Raciocínio Baseado em Casos	82
4.7	Considerações Finais	85
5	Métodos Probabilísticos	87
5.1	Aprendizagem Bayesiana	88
5.2	O Classificador <i>Naive Bayes</i>	91
5.3	Redes Bayesianas para Classificação	96
5.4	Considerações Finais	99
6	Métodos Baseados em Procura	101
6.1	Árvores de Decisão e Regressão	101
6.2	Regras de Decisão	116
6.3	Modelos Avançados para Árvores de Decisão	124
6.4	Considerações Finais	126
7	Métodos Baseados em Otimização	127
7.1	Redes Neurais Artificiais	127
7.2	Máquinas de Vetores de Suporte	144
7.3	Considerações Finais	159
8	Modelos Múltiplos Preditivos	161
8.1	Combinando Previsões de Classificadores	163
8.2	Combinando Classificadores Homogêneos	168
8.3	Combinando Classificadores Heterogêneos	175
8.4	Considerações Finais	183
9	Avaliação de Modelos Preditivos	185
9.1	Métricas de Erro	186
9.2	Amostragem	187
9.3	Problemas de Duas Classes e o Espaço ROC	191
9.4	Testes de Hipóteses	196
9.5	Decomposição Viés-Variância da Taxa de Erro	200
9.6	Considerações Finais	203
III	Modelos Descritivos	205
10	Introdução aos Modelos Descritivos	207

11	Extração de Padrões Frequentes	209
11.1	Extração de Conjuntos de Itens Frequentes	209
11.2	O Algoritmo Apriori	211
11.3	O Algoritmo FP-growth	215
11.4	Sumarização de <i>Itemsets</i>	217
11.5	Considerações Finais	221
12	Análise de Agrupamentos	223
12.1	Definições Básicas	223
12.2	Etapas da Análise de Agrupamentos	229
12.3	Considerações Finais	241
13	Algoritmos de Agrupamentos	243
13.1	Algoritmos Hierárquicos	244
13.2	Algoritmos Particionais Baseados em Erro Quadrático	248
13.3	Algoritmos Baseados em Densidade	250
13.4	Algoritmos Baseados em Grafo	252
13.5	Algoritmos Baseados em Redes Neurais	252
13.6	Algoritmos Baseados em reticulados	253
13.7	Considerações Finais	254
14	Modelos Múltiplos Descritivos	257
14.1	<i>Ensembles</i> de Agrupamentos	258
14.2	Agrupamento Multiobjetivo	270
14.3	<i>Ensemble</i> Multiobjetivo	274
14.4	Considerações Finais	275
15	Avaliação de Modelos Descritivos	277
15.1	Critérios de Validação	278
15.2	Critérios Relativos	283
15.3	Critérios Internos	291
15.4	Critérios Externos	293
15.5	Considerações Finais	298
IV	Tópicos Avançados	299
16	Aprendizagem em Fluxos Contínuos de Dados	303
16.1	Desafios na Aprendizagem em Fluxos Contínuos de Dados	304
16.2	Algoritmos de Aprendizagem em Fluxos de Dados	305
16.3	Deteção de Mudança	310
16.4	Considerações Finais	312

17	Meta aprendizagem	315
17.1	Caraterização de Conjuntos de Dados	317
17.2	Medidas de Avaliação dos Algoritmos	319
17.3	Formas de Apresentação de Sugestões	320
17.4	Recomendação com base na Caraterização	320
17.5	Estudo de Casos	321
17.6	Considerações Finais	322
18	Decomposição de Problemas Multiclasse	323
18.1	Fase de Decomposição	324
18.2	Fase de Reconstrução	332
18.3	Considerações Finais	333
19	Classificação Multirótulo	335
19.1	Principais Abordagens	337
19.2	Densidade e Cardinalidade do Rótulo	341
19.3	Medidas de Avaliação	342
19.4	Considerações Finais	344
20	Classificação Hierárquica	345
20.1	Tipos de Problemas	346
20.2	Algoritmos para Classificação Hierárquica	348
20.3	Avaliação de Classificadores Hierárquicos	352
20.4	Considerações Finais	353
21	Computação Natural	355
21.1	Inteligência de Enxames	356
21.2	Computação Evolutiva	360
21.3	Considerações Finais	364
22	Análise de Redes Sociais	365
22.1	Introdução	365
22.2	Representação de Redes Sociais	368
22.3	Medidas Estatísticas Elementares	370
22.4	Análise de Ligações	378
22.5	Deteção de Comunidades	381
22.6	Propriedades de Redes Reais	387
22.7	Conclusões e Tendências Atuais	391
	Bibliografia	393
	Índice Remissivo	423

Introdução

Em computação, muitos problemas são resolvidos por meio da escrita de um algoritmo que específica, passo a passo, como resolver um problema. No entanto, não é fácil escrever um programa de computador que realize com eficiência algumas tarefas que realizamos com facilidade no nosso dia a dia. Por exemplo, como reconhecer pessoas pelo rosto ou pela fala? Que características dos rostos ou da fala devem ser consideradas? Como podemos codificar aspectos como diferentes expressões faciais de uma mesma pessoa, alterações na face (e.g. óculos, bigode, cortes de cabelo), mudanças na voz (e.g. devido a uma gripe) ou estados de espírito? No entanto, os seres humanos conseguem realizar estas tarefas com relativa facilidade. Fazem isso por meio de reconhecimento de padrões, quando aprendem o que deve ser observado num rosto ou na fala para conseguir identificar pessoas, após terem tido vários exemplos de rostos ou falas com identificação clara.

Um bom médico consegue diagnosticar se um paciente está com problemas cardíacos, tendo por base um conjunto de sintomas e de resultados de exames clínicos. Para esse efeito, o médico utiliza o conhecimento adquirido durante a sua formação e a experiência proveniente do exercício da profissão. Como escrever um programa de computador que, dados os sintomas e os resultados de exames clínicos, apresente um diagnóstico que seja tão bom quanto o de um médico experiente?

Também pode ser difícil escrever um programa que efetue a análise de dados das vendas de uma loja. Para descobrir quantas pessoas fizeram mais de uma compra numa loja no ano anterior, podem ser facilmente utilizados os Sistemas de Gestão de Bases de Dados (SGBD). No entanto, como podemos escrever um programa que responda a questões mais complicadas, como por exemplo:

- Identificar conjuntos de produtos que são frequentemente vendidos em conjunto.
- Recomendar novos produtos a clientes que costumam comprar produtos semelhantes.
- Agrupar os consumidores da loja em grupos de forma a melhorar as operações de marketing.

Não obstante a dificuldade em escrever um programa de computador que possa lidar

de forma eficiente com estas tarefas, o número de vezes em que tarefas tão complexas como estas necessitam ser realizadas diariamente é frequente. Além disso, o volume de informação que precisa ser considerado na sua implementação torna difícil, ou mesmo impossível, a sua realização.

As técnicas de Inteligência Artificial (IA), em particular de Extração de Conhecimento de Dados (ECD) ou Aprendizagem Automática, têm sido utilizadas com sucesso num grande número de problemas reais, incluindo os problemas citados anteriormente.

Este capítulo está organizado da seguinte forma. A Secção 1.1 apresenta a relação entre ECD e IA, mostrando alguns exemplos da utilização de ECD em problemas reais. Na Secção 1.2 é introduzida a relação entre um conjunto de dados e a qualidade da hipótese induzida por um algoritmo de ECD. O conceito de viés indutivo, essencial para que a aprendizagem ocorra, é discutido na Secção 1.3. A Secção 1.4 descreve as diferentes tarefas de aprendizagem, que são agrupadas em aprendizagem preditiva e aprendizagem descritiva. Por fim, a Secção 1.5 apresenta a estrutura dos capítulos do livro.

1.1 Inteligência Artificial e Aprendizagem Automática

Até há alguns anos atrás, a área de IA era vista como uma área teórica, com aplicações apenas em pequenos problemas curiosos, desafiantes, mas de pouco valor prático. Grande parte dos problemas práticos que necessitavam de computação eram resolvidos pela codificação, nalguma linguagem de programação, dos passos necessários à respetiva resolução. A partir da década de 1970, verificou-se uma maior disseminação do uso de técnicas de computação baseadas em IA para a resolução de problemas reais. Muitas vezes, estes problemas eram computacionalmente tratados por meio da aquisição de conhecimento de especialistas do domínio (e.g. especialistas da área da medicina), que era então codificado por regras lógicas, num programa de computador. Estes programas eram conhecidos como Sistemas Especialistas ou Sistemas Baseados em Conhecimento. O processo de aquisição do conhecimento envolvia entrevistas com os especialistas para descobrir as regras utilizadas na tomada de decisão. Esse processo possuía várias limitações, tais como: subjetividade, decorrente do fato dos especialistas sustentarem, com frequência, a tomada de decisão na sua intuição; e dificuldade em verbalizar e exteriorizar esse conhecimento.

Nas últimas décadas, a crescente complexidade dos problemas a serem tratados computacionalmente e o volume de dados gerados em diferentes setores, reforçou a necessidade de desenvolvimento de ferramentas computacionais mais sofisticadas e autónomas, que reduzissem a necessidade de intervenção humana e a dependência de especialistas. Para alcançar estes objetivos, as técnicas desenvolvidas devem ser capazes de criar, de forma autónoma e a partir da experiência passada, uma hipótese, ou função, capaz de resolver o problema que se deseja tratar. Um exemplo simples é a descoberta de uma hipótese, na forma de uma regra ou conjunto de regras, para definir quais os clientes de um supermercado que devem receber publicidade de um novo produto, utilizando os dados de compras anteriores dos clientes registados na base de dados do supermercado. A este processo de indução de uma hipótese (ou aproximação de função) a partir da experiên-

cia passada, dá-se o nome de *Aprendizagem Automática* ou *Extração de Conhecimento de Dados* (ECD).

A capacidade de aprendizagem é considerada essencial para um comportamento inteligente. Atividades como memorizar, observar e explorar situações para aprender fatos, melhorar habilidades motoras/cognitivas através da prática, organizar conhecimento novo e utilizar representações apropriadas, podem ser consideradas atividades relacionadas com aprendizagem. Existem várias definições de ECD na literatura. Uma delas, apresentada em Mitchell (1997), define ECD como:

'A capacidade de melhorar o desempenho na realização de alguma tarefa por meio da experiência.'

Em ECD, os computadores são programados para aprender com a experiência passada. Para tal, empregam um princípio de inferência denominado indução, no qual se obtêm conclusões genéricas a partir de um conjunto particular de exemplos. Assim, os algoritmos de ECD aprendem a induzir uma função, ou hipótese, capaz de resolver um problema, a partir de dados que representam instâncias do problema a ser resolvido. Estes dados formam um conjunto, simplesmente denominado conjunto de dados (Seção 1.2). Embora ECD esteja naturalmente associada à IA, outras áreas de investigação são importantes e têm contribuições diretas e significativas no avanço da ECD, tais como, Probabilidade e Estatística, Teoria da Computação, Neurociência, Teoria da Informação, entre outras. ECD é uma das áreas de investigação da computação que mais tem crescido nos últimos anos. Diferentes algoritmos de ECD, diferentes formas de utilizar os algoritmos existentes e adaptações de algoritmos são continuamente propostos. Além disso, em cada instante surgem novas variações nas características dos problemas reais a serem tratados.

Existem várias aplicações bem-sucedidas de técnicas de ECD na resolução de problemas reais, entre as quais podem ser citadas:

- Reconhecimento de voz - palavras faladas;
- Predição de taxas de cura de pacientes com diferentes doenças;
- Detecção do uso fraudulento de cartões de crédito;
- Condução autónoma de automóveis;
- Ferramentas que jogam gamão e xadrez ao nível de campeões;
- Diagnóstico de cancro por meio da análise de dados de expressão genética.

Além do enorme volume de aplicações que beneficiam das características da área de ECD, outros fatores têm favorecido a expansão desta área, nomeadamente, o desenvolvimento de algoritmos cada vez mais eficazes e eficientes, e a elevada capacidade dos recursos computacionais atualmente disponíveis. Outras motivações para a investigação em ECD incluem a possibilidade de aumentar a compreensão de como se processa a aprendizagem nos seres vivos. Além disso, algumas tarefas são naturalmente melhor definidas por meio de exemplos. Os modelos gerados são, ainda, capazes de lidar com situações não apresentadas durante o seu desenvolvimento, sem necessariamente necessitar de uma nova fase de projeto.

1.2 Indução de Hipóteses

Para ilustrar a relação entre ECD e indução de hipóteses, Neste conjunto, que será denominado *hospital*, cada observação (também denominada objeto, exemplo, padrão ou registro) corresponde a um doente. Cada observação é uma tupla formada pelos valores das características, ou atributos, referentes ao paciente, que descrevem os seus principais aspetos. Os atributos (também denominados campos ou variáveis) utilizados para cada paciente podem ser, por exemplo, a sua identificação, o seu nome, a sua idade, o seu sexo, o seu estado de origem e os sintomas e resultados de exames clínicos. Exemplos de sintomas podem ser presença e distribuição de manchas na pele, peso e temperatura do corpo.

Conforme será visto mais adiante, em algumas tarefas de extração de conhecimento, um dos atributos é considerado o atributo de saída (também denominado atributo alvo ou atributo meta), cujos valores podem ser estimados utilizando os valores dos demais atributos, denominados atributos de entrada, ou atributos previsoires. O objetivo de um algoritmo de ECD utilizado nestas tarefas é aprender, a partir de um subconjunto dos dados, denominado conjunto de treino, um modelo ou hipótese capaz de relacionar os valores dos atributos de entrada de um objeto com o valor do seu atributo de saída.

Um requisito importante para os algoritmos de ECD é a capacidade de lidar com dados imperfeitos. Muitos conjuntos de dados apresentam algum tipo de problema, como presença de ruído, dados inconsistentes, dados em falta e dados redundantes. Idealmente, os algoritmos de ECD devem ser robustos a estes problemas, minimizando a sua influência no processo de indução de hipóteses. Porém, dependendo da sua extensão, estes problemas podem prejudicar o processo indutivo. Por esse motivo, técnicas de pré-processamento são frequentemente utilizadas na identificação e minimização da ocorrência desses problemas.

Retomando o exemplo dos pacientes, considere a situação em que um algoritmo de ECD é utilizado para aprender uma hipótese (por exemplo, uma regra) capaz de diagnosticar pacientes de acordo com os valores associados aos seus atributos de entrada. Os atributos referentes à identificação e nome do paciente não são considerados entradas relevantes, uma vez que não possuem qualquer tipo de relação com o diagnóstico de uma doença. Na verdade, o que se pretende, é induzir uma hipótese capaz de realizar diagnósticos corretos para novos pacientes, i.e. pacientes diferentes daqueles que foram utilizados na aprendizagem da regra de decisão. Assim, uma vez induzida uma hipótese, é desejável que esta também seja válida para outros objetos do mesmo domínio, ou problema, que não fazem parte do conjunto de treino. A esta propriedade de uma hipótese continuar a ser válida para novos objetos dá-se o nome de *capacidade de generalização da hipótese*. Para que uma hipótese se revista de utilidade quando aplicada a novos dados, é fundamental que apresente uma boa capacidade de generalização.

Quando uma hipótese apresenta uma capacidade de generalização reduzida, pode ser porque esta se encontra superajustada aos dados (*overfitting*). Neste caso, diz-se que a hipótese memorizou, ou especializou-se nos dados de treino. No caso inverso, o algoritmo de ECD pode induzir hipóteses que apresentem uma baixa taxa de acerto mesmo no conjunto de treino, gerando uma condição de subajustamento (*underfitting*). Esta situação

pode ocorrer, por exemplo, quando os exemplos de treino disponíveis são pouco representativos, ou o modelo usado é muito simples e, por conseguinte, incapaz de capturar os padrões existentes nos dados (Monard e Baranauskas, 2003). Estes conceitos são ilustrados e novamente discutidos na Secção 7.2.1. São feitas então considerações e motivações sobre a escolha de modelos com boa capacidade de generalização.

1.3 Viés Indutivo

Quando um algoritmo de ECD aprende a partir de um conjunto de dados de treino, procura uma hipótese, no espaço de hipóteses possíveis, que seja capaz de descrever as relações entre os objetos, e que melhor se ajuste aos dados de treino.

Cada algoritmo utiliza uma forma, ou representação, para descrever a hipótese induzida. Por exemplo, as redes neuronais artificiais representam uma hipótese por um conjunto de valores reais, associados aos pesos das conexões da rede. As árvores de decisão utilizam uma estrutura de árvore em que cada nó interno é representado por uma pergunta referente ao valor de um atributo e cada nó externo está associado a uma classe. A representação utilizada define a preferência ou viés (*bias*) de representação do algoritmo e pode restringir o conjunto de hipóteses que podem ser induzidas pelo algoritmo. A Figura 1.1 ilustra o viés de representação utilizado por técnicas de indução de árvores de decisão, redes neuronais artificiais e regras de decisão.

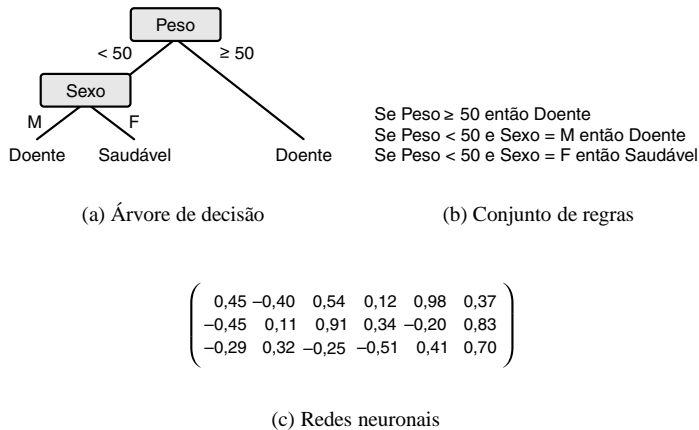


Figura 1.1 Diferentes vieses de representação.

Além do viés de representação, os algoritmos de ECD possuem também um viés de procura. O viés de procura de um algoritmo é a forma como o algoritmo procura a hipótese que melhor se ajusta aos dados de treino. Este viés define como as hipóteses são procu-

radar no espaço de hipóteses. Por exemplo, o algoritmo ID3, que é utilizado na indução de árvores de decisão, tem como viés de procura a preferência por árvores de decisão com poucos nós, conforme será apresentado no Capítulo 6.

Assim, cada algoritmo de ECD possui dois vieses: um *viés de representação* e um *viés de procura*. O viés é necessário para restringir as hipóteses a serem visitadas no espaço de procura. Sem viés não haveria aprendizagem/generalização. Os modelos seriam especializados para os exemplos individuais. Embora, à primeira vista, o viés pareça ser uma limitação dos algoritmos de ECD, segundo Mitchell (1997), sem viés um algoritmo de ECD não consegue generalizar o conhecimento adquirido durante o treino para aplicá-lo com sucesso a novos dados.

1.4 Tarefas de aprendizagem

Os algoritmos de ECD têm sido amplamente utilizados em diversas tarefas, que podem ser organizadas de acordo com diferentes critérios. Um deles diz respeito ao paradigma de aprendizagem a ser adotado para lidar com a tarefa. De acordo com esse critério, as tarefas de aprendizagem podem ser divididas em: **Preditivas** e **Descritivas**.

Em tarefas de previsão, o objetivo consiste em encontrar uma função (também denominada de modelo, ou hipótese), a partir dos dados de treino, que possa ser utilizada para prever um rótulo, ou valor, que caracterize um novo exemplo, com base nos valores dos seus atributos de entrada. Para esse efeito, é necessário que cada objeto do conjunto de treino possua atributos de entrada e de saída.

Os algoritmos, ou métodos, de ECD utilizados nesta tarefa induzem modelos preditivos. Estes algoritmos seguem o paradigma da aprendizagem supervisionada. O termo *supervisionada* vem da simulação da presença de um *supervisor externo*, que conhece a saída (rótulo) associada a cada exemplo (conjunto de valores para os atributos de entrada). Com base neste conhecimento, o supervisor externo pode avaliar a capacidade da hipótese induzida em prever o valor de saída para novos exemplos.

Em tarefas de descrição, o objetivo consiste em explorar, ou descrever, um conjunto de dados. Os algoritmos de ECD utilizados nestas tarefas ignoram o atributo de saída. Por esse motivo, diz-se que estes algoritmos seguem o paradigma de aprendizagem não supervisionada. Por exemplo, uma tarefa descritiva de agrupamento de dados tem por meta encontrar grupos de objetos semelhantes no conjunto de dados. Outra tarefa descritiva consiste em encontrar regras de associação que relacionam um grupo de atributos com outro grupo de atributos.

A Figura 1.2 apresenta uma hierarquia de aprendizagem, de acordo com os tipos de tarefas de aprendizagem. No topo aparece a aprendizagem indutiva, processo pelo qual são realizadas as generalizações a partir dos dados. Em seguida, surgem os tipos de aprendizagem supervisionada (preditivo) e não supervisionada (descritivo). As tarefas supervisionadas distinguem-se pelo tipo dos rótulos dos dados: discreto, no caso de classificação; e contínuo, no caso de regressão. As tarefas descritivas são genericamente divididas em: agrupamento, em que os dados são agrupados de acordo com sua semelhança; sumari-

zação, cujo objetivo é encontrar uma descrição simples e compacta de um conjunto de dados; e associação, que consiste em encontrar padrões frequentes de associações entre os atributos de um conjunto de dados. Com exceção da sumarização, as demais tarefas serão descritas neste livro.

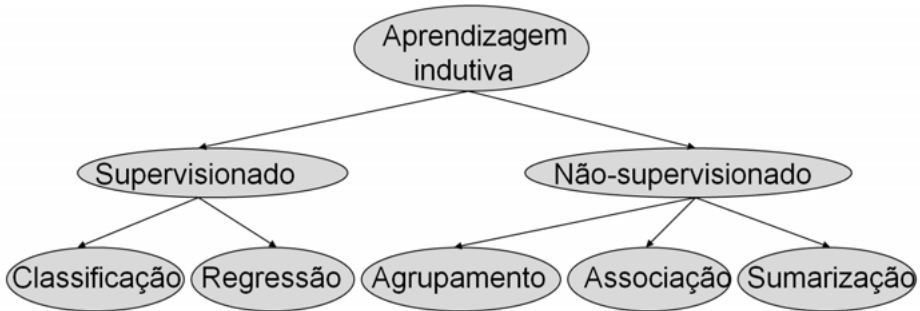


Figura 1.2 Hierarquia de aprendizagem.

Note-se que, apesar desta divisão básica de modelos em preditivos e descritivos, um modelo preditivo também providencia uma descrição compacta de um conjunto de dados, e um modelo descritivo pode efetuar previsões após ser validado.

Uma tarefa de aprendizagem que não se enquadra nas tarefas anteriores é a de *aprendizagem por reforço*. Nesta tarefa, o objetivo é reforçar, ou recompensar, uma ação considerada positiva, e punir uma ação considerada negativa. Um exemplo de tarefa de reforço é ensinar um robô a encontrar a melhor trajetória entre dois pontos. Os algoritmos de aprendizagem utilizados nesta tarefa, em geral, punem a passagem por caminhos pouco promissores, e recompensam a passagem por caminhos promissores. Devido ao foco adotado para este livro, esta tarefa não será abordada.

1.5 Estrutura do Livro

Este livro tem por objetivo apresentar os principais conceitos e algoritmos de ECD e mostrar como ECD pode ser utilizada para na resolução de problemas reais. Para esse efeito, serão cobertos tanto temas tradicionais como resultados de pesquisas recentes na área.

De forma a agrupar os temas cobertos de uma maneira mais uniforme, os capítulos do livro foram organizados em três grandes temas ou módulos:

- **Preparação de dados:** engloba tópicos de descrição dos dados, análise estatística de dados e pré-processamento de dados.
- **Métodos preditivos:** este módulo está relacionado com o paradigma da aprendizagem supervisionada e, após definir os conceitos gerais referentes a este tema,

descreve os principais algoritmos de aprendizagem preditiva, explica como as hipóteses podem ser combinadas formando comités, introduz possíveis estratégias para planejar experiências com esses métodos, e descreve as principais métricas utilizadas na avaliação do seu desempenho.

- **Modelos descritivos:** este módulo foca a aprendizagem não supervisionada. São abordados os temas de padrões frequentes e análise de agrupamentos. Descrevemos os conceitos básicos, os algoritmos principais, e as formas de combinação. É também discutido como as experiências utilizando estes métodos podem ser planejados e avaliados.
- **Tópicos avançados:** inclui temas de investigação recente na área de ECD. Os temas considerados são: fluxos de dados, meta-aprendizagem, estratégias para classificação multiclasse, classificação hierárquica, classificação multirótulo e análise de redes sociais.

Estes tópicos foram cuidadosamente escolhidos, de modo a que os leitores tenham acesso a uma dose equilibrada entre abrangência e profundidade dos temas básicos e avançados nas áreas de Inteligência Artificial, que utilizam aprendizagem automática na indução de modelos de decisão. Esperamos que este livro, ao mesmo tempo que introduz o leitor aos principais aspectos de ECD e a temas de investigação recentes, sirva de alicerce à realização de investigação que promova o crescimento e o fortalecimento da área. Esperamos ainda que o livro estimule o leitor a utilizar as várias técnicas aqui abordadas na resolução de problemas reais.

Parte I

Preparação de Dados

Introdução

Todos os dias é gerada uma enorme quantidade de dados. Estima-se que, a cada 20 meses, a quantidade de dados armazenada em todas as bases de dados do mundo duplica (Witten et al., 2011). Estes dados são gerados por atividades como transações financeiras, monitorização ambiental, obtenção de dados clínicos e genéticos, captura de imagens, tráfego na internet, entre outras. Os dados podem, ainda, assumir vários formatos diferentes, como séries temporais, *itemsets*, transações, grafos ou redes sociais, textos, páginas web, imagens (vídeos) e áudio (músicas). Com o aumento crescente da quantidade de dados gerada, o fosso entre a quantidade de dados existente e a porção de dados que é analisada e compreendida tem-se acentuado de forma significativa ao longo do tempo.

Conjuntos de dados são formados por objetos que podem representar um objeto físico, como uma cadeira, ou uma noção abstrata, como os sintomas apresentados por um paciente que se dirige a um hospital. Em geral, cada objeto é descrito por um conjunto de atributos de entrada, ou vetor de características. Cada objeto corresponde a uma ocorrência dos dados. Cada atributo está associado a uma propriedade do objeto.

Formalmente, um conjunto de dados pode ser representado por uma matriz de objetos $\mathbf{X}_{n \times d}$, em que n é o número de objetos e d é o número de atributos de entrada de cada objeto. O valor de d define a dimensionalidade dos objetos, ou do espaço de objetos (também denominado espaço de entradas, ou espaço de atributos). Cada elemento dessa matriz, x_i^j , contém o valor da j -ésima característica para o i -ésimo objeto. Os d atributos também podem ser vistos como um conjunto de eixos ortogonais e os objetos, como pontos no espaço de dimensão d , também designado por espaço de objetos. A Figura 1.3 ilustra um exemplo de um espaço de objetos. Nesse espaço, a posição de cada objeto é definida pelos valores de dois atributos de entrada ($d = 2$) que, neste caso, representam os resultados de dois exames clínicos. O atributo de saída é representado pelo formato do objeto na figura: círculo para pacientes doentes e triângulo para pacientes saudáveis.

Apesar do crescente número de bases de dados disponíveis, na maioria das vezes não é possível utilizar diretamente algoritmos de ECD sobre esses dados. Técnicas de pré-processamento são frequentemente utilizadas para tornar os conjuntos de dados mais adequados para o uso de algoritmos de ECD. Essas técnicas podem ser agrupadas nas seguintes tarefas: **Integração de dados**, **Amostragem de dados**, **Balanceamento de dados**, **Limpeza de dados**, **Redução de dimensionalidade**, e **Transformação de dados**.

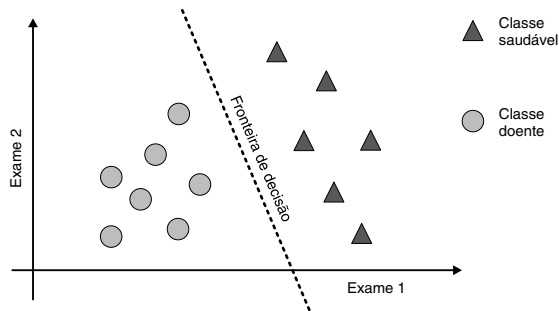


Figura 1.3 Objetos no espaço definido pelos atributos.

Grande parte das empresas, órgãos do governo e outras organizações possuem os seus dados armazenados em bases de dados. Assim, os dados podem ser oriundos de mais do que uma fonte, ou tabela atributo-valor. Quando os dados presentes em diferentes conjuntos precisam ser utilizados por um algoritmo de ECD, esses conjuntos devem ser integrados de forma a constituir uma única tabela. Essa integração pode levar a inconsistências e redundâncias. Os algoritmos de ECD também podem apresentar dificuldades quando precisam lidar com um grande volume de dados. Essa grande quantidade pode estar relacionada com o número de objetos, com o número de atributos, ou ambos. Problemas como a redundância e a inconsistência, estão muitas vezes relacionados com a quantidade de dados. Técnicas de amostragem e de seleção de atributos têm sido empregues para amenizar estes problemas. Em dados reais, a distribuição dos objetos entre as classes pode não ser uniforme. Por conseguinte, algumas classes podem ter um número de objetos muito superior a outras, formando um conjunto de dados desbalanceado. Alguns algoritmos de ECD têm dificuldade em induzir um bom modelo a partir de conjuntos desbalanceados. Muitos dos conjuntos de dados reais apresentam problemas, tais como, a presença de ruído e dados incompletos e/ou inconsistentes. Os dados podem estar incompletos devido à ausência de valores. Os dados podem ser inconsistentes por causa de erros na sua geração, captação ou entrada. O desempenho da maioria dos algoritmos de ECD é afetado pela presença destes problemas. Para lidar com eles, diversas técnicas para limpeza de dados têm sido propostas e investigadas na literatura de ECD. Mesmo após a eliminação de atributos por especialistas no domínio, os atributos que restam podem dificultar a tarefa de algoritmos de ECD, devido a diversos motivos, como sejam a presença de um número muito grande de atributos, de atributos redundantes, irrelevantes e/ou inconsistentes.

Vários algoritmos de ECD têm dificuldade em utilizar os dados no seu formato original. Para tratar este problema, são efetuadas transformações aos dados originais antes destes serem utilizados pelo algoritmo. Um exemplo de transformação é a conversão de valores simbólicos em valores numéricos.

Análise Exploratória de Dados

A análise das características presentes num conjunto de dados permite a descoberta de padrões e tendências que podem fornecer informações valiosas que ajudem a compreender o processo que gera os dados. Muitas dessas características podem ser obtidas por meio da aplicação de fórmulas estatísticas simples. Outras podem ser observadas por meio do uso de técnicas de visualização. Neste capítulo são descritas as principais características para a análise e compreensão de um conjunto de dados utilizados em experiências de ECD. Analisa-se a forma como os dados podem estar organizados e os tipos de valores que estes podem assumir. Serão apresentados ainda vários tipos de gráficos que facilitam a análise visual da distribuição dos valores em conjuntos de dados com uma ou mais variáveis. Este capítulo está organizado da seguinte maneira. A Seção 2.1 descreve como os atributos de um conjunto de dados podem ser caracterizados pelo seu tipo e escala. Por fim, na Seção 2.2, são apresentadas várias medidas, assim como gráficos, que permitem descrever conjuntos de dados, tanto univariados como multivariados.

2.1 Caracterização de Dados

Os conjuntos de dados são formados por objetos que podem representar um objeto físico, como uma cadeira, ou uma noção abstrata, como os sintomas apresentados por um paciente que se dirige a um hospital. Estes objetos são também designados por instâncias, objetos, registros ou exemplos. Em geral, são representados por um vetor de características que os descreve, também denominadas atributos, campos ou variáveis. Cada objeto corresponde a uma linha na tabela de dados. Cada atributo está associado a uma propriedade do objeto.

Formalmente, os dados podem ser representados por uma matriz de objetos $\mathbf{X}_{n \times d}$, em que n é o número de objetos e d é o número de atributos de entrada de cada objeto. O valor de d define a dimensionalidade dos objetos ou do espaço de objetos. Cada elemento dessa matriz, x_i^j , contém o valor da j -ésima característica para o i -ésimo objeto. Os d atributos também podem ser interpretados como um conjunto de eixos ortogonais, e os objetos como pontos no espaço de dimensão d , denominado espaço de objetos.

Para ilustrar os conceitos abordados neste capítulo com um exemplo prático, considere

novamente o conjunto de dados provenientes de pacientes de um hospital, denominado *hospital*, ilustrado pela Tabela 2.1. Este conjunto foi inicialmente apresentado no Capítulo 1. No conjunto *hospital*, cada objeto corresponde a um paciente, sendo por isso formado pelos valores de atributos de entrada (também denominados atributos preditivos) referentes ao paciente. Esses atributos são: identificação, nome, idade, sexo, sintomas e resultados de exames clínicos. Exemplos de sintomas são presença e distribuição de manchas na pele, peso do paciente e temperatura do seu corpo. Além destes atributos, a tabela apresenta um atributo alvo, também denominado atributo meta ou de saída, que representa o fenômeno de interesse sobre o qual se pretende fazer previsões. Em tarefas descritivas, os dados não apresentam este atributo e, muitas vezes, a definição deste tipo de atributo pode ser obtida como um dos seus resultados. Por outro lado, as tarefas preditivas baseiam-se na presença deste atributo, como mencionado no Capítulo 1. Na maioria dos casos, os dados apresentam apenas um atributo alvo ¹.

Quando um atributo alvo contém rótulos que identificam categorias ou classes às quais os objetos pertencem, é denominado classe e assume valores discretos $1, \dots, k$. Nestes casos, estamos perante um problema de classificação. Se o número de objetos por classe for diferente, a classe mais frequente é denominada classe maioritária, e a menos frequente, minoritária. Por outro lado, se o atributo alvo é descrito por valores numéricos contínuos, estamos perante um problema de regressão (Mitchell, 1997). Um caso especial de regressão é a previsão de valores em séries temporais, que se caracteriza pelo fato de os seus valores apresentarem uma relação de periodicidade. Quer em problemas de classificação quer em problemas de regressão, os restantes atributos são denominados atributos preditivos, por poderem ser utilizados na previsão do valor do atributo alvo.

Na Tabela 2.1, para cada paciente são apresentados os valores dos atributos *Id.* (identificação do paciente), *Nome*, *Idade*, *Sexo*, *Peso*, *Manchas* (presença e distribuição de manchas no corpo), *Temp.* (temperatura do corpo), *#Int.* (número de internamentos), *Est.* (estado de origem) e *Diagnóstico*, que indica o diagnóstico do paciente e corresponde ao atributo alvo. As tabelas com este formato também são denominadas por tabelas atributo-valor.

O domínio de um atributo, ou seja os possíveis valores que um atributo pode assumir, determina o tipo de análise que podemos efetuar. Neste livro, consideramos dois aspetos: *tipo* e *escala*. O tipo de um atributo diz respeito ao grau de quantificação nos dados, e a escala indica a significância relativa dos seus valores. Conhecer o tipo e a escala dos atributos permite identificar a forma adequada para a preparação dos dados e posterior modelação. As definições que apresentamos são utilizadas para classificar os valores que os atributos podem assumir no que diz respeito aos dois aspetos mencionados (Jain e Dubes, 1988; Barbara, 2000; Yang et al., 2005).

¹ Resultados mais recentes consideram dados com mais de um atributo alvo. Este é o foco, por exemplo, da classificação multirótulo.

Tabela 2.1 Conjunto de dados *hospital* com seus atributos

Id.	Nome	Idade	Sexo	Peso	Manchas	Temp.	# Int.	Est.	Diagnóstico
4201	João	28	M	79	Concentradas	38,0	2	SP	Doente
3217	Maria	18	F	67	Inexistentes	39,5	4	MG	Doente
4039	Luiz	49	M	92	Espalhadas	38,0	2	RS	Saudável
1920	José	18	M	43	Inexistentes	38,5	8	MG	Doente
4340	Cláudia	21	F	52	Uniformes	37,6	1	PE	Saudável
2301	Ana	22	F	72	Inexistentes	38,0	3	RJ	Doente
1322	Marta	19	F	87	Espalhadas	39,0	6	ECD	Doente
3027	Paulo	34	M	67	Uniformes	38,4	2	GO	Saudável

2.1.1 Tipo

O tipo define se o atributo representa quantidades, sendo então denominado quantitativo ou numérico, ou qualidades, sendo então designado de qualitativo, simbólico ou categórico, pois os seus valores podem ser associados a categorias. Exemplos de conjuntos de valores qualitativos são $\{\text{pequeno, médio, grande}\}$ e $\{\text{matemática, física, química}\}$. Apesar de alguns atributos qualitativos poderem ter os respectivos valores ordenados, não podem ser aplicadas operações aritméticas. Os atributos quantitativos são numéricos, como no conjunto de valores $\{23, 45, 12\}$. Os valores de um atributo quantitativo são ordenados e podem ser utilizados em operações aritméticas. Os valores quantitativos podem ser ainda contínuos ou discretos.

Os atributos contínuos podem assumir um número infinito de valores. Geralmente esses atributos são resultados de medidas, sendo os seus valores representados por números reais. No entanto, deve-se tomar em consideração que em computadores digitais a precisão para valores reais é, geralmente, limitada. Exemplos de atributos contínuos são atributos que representam pesos, tamanhos ou distâncias.

Os atributos discretos contêm um número finito ou infinito contável de valores. Um caso especial de atributos discretos são os atributos binários (ou booleanos), que apresentam apenas dois valores, como 0/1, sim/não, ausência/presença e verdadeiro/falso. Para efeitos ilustrativos, na Tabela 2.2 é apresentada a classificação por tipo dos atributos presentes no conjunto de dados da Tabela 2.1.

É importante observar que uma medida quantitativa possui, além do valor numérico, uma unidade, por exemplo, *metro*. No processo de extração de conhecimento, se o atributo *altura* assume o valor 100, o valor em si não indica se a altura é medida em centímetros, metros ou jardas, e essa informação pode ser importante na avaliação do conhecimento adquirido.

Os atributos quantitativos ou numéricos podem assumir valores binários, inteiros ou reais. Por outro lado, atributos qualitativos são, geralmente, representados por um número finito de símbolos ou nomes. Em alguns casos, os atributos categóricos são representados por números. No entanto, estes números não representam quantidades pelo que não são passíveis de serem submetidos a operações aritméticas. Por exemplo, qual seria o sentido

Tabela 2.2 Tipo dos atributos do conjunto *hospital*

Atributo	Classificação
Id.	Qualitativo
Nome	Qualitativo
Idade	Quantitativo discreto
Sexo	Qualitativo
Peso	Quantitativo contínuo
Manchas	Qualitativo
Temp.	Quantitativo contínuo
#Int.	Quantitativo discreto
Est.	Qualitativo
Diagnóstico	Qualitativo

de calcular a média dos valores de um atributo categórico representando o número de identificação de um paciente?

2.1.2 Escala

A escala define as operações que podem ser realizadas sobre os valores do atributo. Em relação à escala, os atributos podem ser classificados como nominais, ordinais, intervalares e racionais. Os dois primeiros são do tipo qualitativo e os dois últimos são do tipo quantitativo. Estas quatro escalas são seguidamente definidas em detalhe.

Na escala nominal, os valores consistem apenas nomes diferentes, carregando a menor quantidade de informação possível, não existindo uma relação de ordem entre os seus valores. Consequentemente, as operações mais utilizadas na manipulação dos seus valores são as de igualdade e desigualdade entre valores. Por exemplo, se o atributo representa continentes do planeta, apenas é possível verificar se dois valores são iguais ou diferentes (a não ser que se pretenda ordenar os continentes por ordem alfabética, mas nesse caso o atributo seria do tipo ordinal). São exemplos de atributos com escala nominal: nome do paciente, RG, CPF, número da conta no banco, CEP, cores (com as categorias verde, amarelo, branco etc.) e sexo (com as categorias feminino e masculino).

Os valores numa escala ordinal refletem também uma ordem das categorias representadas. Dessa forma, além dos operadores anteriores, podem também ser utilizados operadores como $<$, $>$, \leq , \geq . Por exemplo, quando um atributo categórico possui como domínio o conjunto *pequeno*, *médio* e *grande*, é possível definir uma relação de ordem, ou seja indicar se um valor é igual, maior ou menor que outro. Exemplos de atributos com escala ordinal incluem a hierarquia militar e avaliações qualitativas de temperatura, como frio, morno e quente.

Na escala intervalar, os atributos são representados por números que variam dentro de um intervalo. Assim, é possível definir tanto a ordem como a diferença em magnitude entre dois valores. A diferença em magnitude indica a distância que separa dois valores no

Tabela 2.3 Escala dos atributos do conjunto *hospital*

Atributo	Classificação
Id.	Nominal
Nome	Nominal
Idade	Racional
Sexo	Nominal
Peso	Racional
Manchas	Nominal
Temp.	Intervalar
#Int.	Racional
Est.	Nominal
Diagnóstico	Nominal

intervalo de possíveis valores. O valor zero não tem o mesmo significado que o zero utilizado em operações aritméticas. Por exemplo, considere a escala de temperatura medida em graus Celsius. Se o instituto de previsão do tempo em dias de verão, para uma dada cidade, informa que a temperatura vai variar entre 26 e 34 graus e em dias de inverno, para a mesma cidade, informa que vai variar entre 13 e 21 graus, temos que a temperatura dessa cidade apresenta uma variação de 8 graus entre a mínima e a máxima, independente da estação do ano. por outro lado, 90 graus Celsius é diferente de 90 graus Fahrenheit, apesar de ambos os valores se referirem ao atributo temperatura, do mesmo modo que não é possível afirmar que a cidade é duas vezes mais quente no verão do que no inverno. Não faz sentido, para este tipo de atributo, utilizar como informação o quociente entre dois valores. Isto ocorre porque o ponto em que o atributo assume o valor 0, denominado de ponto zero ou origem da escala, é definido de forma arbitrária. Esta é uma característica dos atributos intervalares. Este problema seria eliminado caso fosse utilizada a escala de temperatura Kelvin, cujo valor do ponto zero é o ponto zero verdadeiro (Pyle, 1999). Outros exemplos são a duração de um evento em minutos e datas num calendário.

Os atributos com escala racional são os que contêm mais informação. Os números têm um significado absoluto, ou seja, existe um zero absoluto juntamente com uma unidade de medida que confere significado ao quociente. Por exemplo, considerando o número de vezes que uma pessoa foi ao hospital, o ponto zero está associado à ausência de visitas. Se um paciente foi internado duas vezes e outro foi internado oito vezes, é correto afirmar que o segundo paciente esteve internado quatro vezes mais que o primeiro. Ou seja, faz sentido, para estes atributos, utilizar a razão entre dois valores. Outros exemplos de atributos com escala de razão incluem o tamanho, a distância e os valores financeiros, tais como salário ou saldo em conta-corrente.

Tendo por base a descrição anterior, os atributos do conjunto de dados da Tabela 2.1 podem ser classificados como indicado na Tabela 2.3.

Recentemente, têm sido adotados tipos mais complexos de atributos, como os atributos

hierárquicos, cujos valores representam uma hierarquia de valores, ou atributos simbólicos, cujos valores podem assumir intervalos ou histogramas.

2.2 Exploração de Dados

A informação obtida na análise exploratória de dados pode ajudar, por exemplo, na seleção das técnicas mais apropriadas para o pré-processamento dos dados assim como para a construção de modelos preditivos. Uma das formas mais simples de explorar um conjunto de dados consiste em extrair estatísticas descritivas. A estatística descritiva resume de forma quantitativa as principais características de um conjunto de dados. Muitas dessas medidas são calculadas rapidamente. Por exemplo, no conjunto de dados de pacientes, duas medidas estatísticas podem ser facilmente calculadas: a idade média dos pacientes e a percentagem de pacientes do sexo masculino.

A estatística descritiva assume que os dados são gerados por um processo estatístico. Como o processo pode ser caracterizado por vários parâmetros, as medidas podem ser interpretadas como estimativas dos parâmetros estatísticos da distribuição que gerou os dados. Por exemplo, os dados podem ter sido gerados por uma distribuição normal com média 0 e variância 1. Estas medidas permitem capturar informações como:

- Frequência;
- Localização ou tendência central (por exemplo, a média);
- Dispersão (por exemplo, o desvio padrão);
- Distribuição ou formato.

A medida mais simples é a frequência, que mede a proporção de vezes que um atributo assume um dado valor num determinado conjunto de dados. Pode ser aplicada quer a valores numéricos quer a valores simbólicos. Um exemplo da sua utilização seria: *Num conjunto de dados médicos, 40% dos pacientes têm febre.*

As outras medidas diferem nos casos em que os dados apresentam apenas um atributo (dados univariados) ou mais do que um atributo (dados multivariados), sendo geralmente aplicadas a dados numéricos. Apesar da maioria dos conjuntos de dados utilizados em ECD apresentar mais de um atributo, as análises realizadas para cada atributo podem oferecer informações valiosas sobre os dados.

Seguidamente serão apresentadas as principais medidas de centralidade, dispersão e distribuição para dados uni e multivariados.

2.2.1 Dados Univariados

Para os dados univariados, um objeto x_i é descrito por apenas um atributo. Um conjunto com n objetos pode ser então representado por $\mathbf{x}^j = \{x_1, x_2, \dots, x_n\}$, em que cada x_i é representado por um único valor. É importante observar que o termo *conjunto* não tem o

mesmo significado do termo utilizado em teoria dos conjuntos. Num conjunto de dados, o mesmo valor pode surgir mais do que uma vez num mesmo atributo.

Medidas de Centralidade

As medidas de centralidade definem pontos de referência nos dados e variam para dados numéricos e simbólicos. Para dados simbólicos, geralmente utiliza-se a moda, que corresponde ao valor encontrado com maior frequência para um dado atributo. Por exemplo, a moda do atributo manchas na Tabela 2.1 é o valor *Inexistentes*, que aparece em três dos oito pacientes.

Para os atributos numéricos, as medidas mais utilizadas são a média, a mediana e o percentil. Suponha um conjunto de n valores numéricos: $\mathbf{x}^j = \{x_1, x_2, \dots, x_n\}$. O valor médio desse conjunto pode ser facilmente calculado pela Equação 2.1.

$$\bar{x}^j = \frac{1}{n} \sum_{i=1}^n x_i \quad (2.1)$$

Um dos problemas associados à média é a sua sensibilidade à presença de *outliers*, que são valores muito diferentes dos demais valores observados para o mesmo atributo (ver detalhes no Capítulo 3). A média é um bom indicador do valor central de um conjunto de valores apenas se os valores estão distribuídos de forma simétrica. Um valor muito mais alto ou muito mais baixo que os demais valores do conjunto pode gerar um valor da média distorcido, o que poderia mudar radicalmente se o *outlier* for retirado. Este problema é minimizado com o uso da mediana, que é menos sensível a *outliers*. Para utilizar a mediana, o primeiro passo consiste em ordenar de forma crescente o conjunto de valores. Ordenados os valores, a Equação 2.2 pode ser utilizada para o cálculo da mediana.

$$\text{mediana}(\mathbf{x}^j) = \begin{cases} \frac{1}{2}(x_r + x_{r+1}) & \text{se } n \text{ for par } (n = 2r) \\ x_{r+1} & \text{se } n \text{ for ímpar } (n = 2r + 1) \end{cases} \quad (2.2)$$

Assim, se o número de valores, n , é ímpar, a mediana é igual ao valor do meio do conjunto ordenado. Caso contrário, se for par, é dada pela média dos dois valores do meio. Por exemplo, seja o conjunto de valores $\{17, 4, 8, 21, 4\}$. A ordenação desse conjunto gera a sequência de valores $(4, 4, 8, 17, 21)$. Observe que valores repetidos são mantidos na sequência. Como o número de valores é ímpar, 5, a mediana é dada pelo terceiro valor, isto é, mediana = 8. Se o conjunto de valores fosse formado pelos elementos, $\{17, 4, 8, 21, 4, 15, 13, 9\}$, como o número de elementos, 8, é par, a mediana é dada pela média entre o quarto e o quinto valor da sequência ordenada $(4, 4, 8, 9, 13, 15, 17, 21)$. Nesse caso, mediana = $(9 + 13)/2 = 11$. O recurso à mediana facilita a observação da assimetria da distribuição e da existência de *outliers*. Existem ainda variações da média e da mediana, como, por exemplo, a média truncada, que minimiza os problemas da média por meio do descarte dos exemplos nos extremos da sequência ordenada dos valores. Para isso, é necessário definir a percentagem de exemplos a serem eliminados em cada extremidade.

Algoritmo 2.1 Algoritmo para cálculo do percentil

Entrada: Um conjunto de n valores e o percentil p (valor real entre 0,0 (equivale a 0%) e 1,0 (equivale a 100%)) a ser retornado

Saída: Valor do percentil

- 1 Ordenar os n valores em ordem crescente
 - 2 Calcular o produto np
 - 3 **se** np não for um número inteiro **então**
 - 4 Arredondar para o próximo inteiro
 - 5 Retornar o valor dessa posição na sequência
 - 6 **fim**
 - 7 **senão**
 - 8 Considerar $np = k$
 - 9 Retornar a média entre os valores nas posições k e $k + 1$
 - 10 **fim**
-

Outras medidas muito utilizadas são os quartis e os percentis. Assim como a mediana, estas medidas são utilizadas após prévia ordenação dos valores. Enquanto a mediana divide os dados ao meio, estas outras medidas utilizam pontos de divisão diferentes. Os quartis dividem os valores ordenados em quartos. Assim, o 1º quartil de uma sequência, $Q1$, é o valor para o qual 25% dos outros valores são inferiores a ele. Este também é o valor do 25º percentil, $P25$. O 2º quartil é a mediana, que é equivalente ao 50º percentil, $P50$.

Seja p um valor entre 0 e 100. O p º percentil, Pp , é um valor x_i do conjunto de valores tal que $p\%$ dos valores observados são menores que x_i . Assim, o 40º percentil, $P40$, de um conjunto de valores é o valor para o qual 40% dos demais valores são menores ou iguais a ele. Para calcular o p º percentil, basta seguir os passos do Algoritmo 2.1.

Uma forma simples de visualizar a distribuição dos dados consiste em utilizar técnicas de visualização como, por exemplo, *boxplots*, histogramas e *scatter plots*, que serão introduzidos posteriormente neste capítulo. Para efeitos de ilustrar algumas das medidas e gráficos, será utilizado o conjunto de dados *iris* (Fisher, 1936). O conjunto de dados *iris* é um dos mais populares em ECD, contendo 150 objetos, cada um deles caracterizado por quatro atributos de entrada (tamanho da sépala, largura da sépala, tamanho da pétala e largura da pétala), que assumem valores contínuos, e um atributo alvo com três valores nominais ou classes, que representam as espécies íris setosa, íris versicolor e íris virgínica.

A Figura 2.1 mostra duas variações do gráfico *boxplot* para o atributo largura da sépala, do conjunto de dados *iris*. O *boxplot*, também designado por diagrama de Box e Whisker ou *caixa de bigodes*, apresenta um resumo dos valores para o 1º, 2º (mediana) e 3º quartis, além dos limites inferior e superior.

Do lado esquerdo da figura é apresentado o gráfico *boxplot* original, onde a linha ho-

João Gama é investigador no LIAAD-INESC TEC e Professor Associado na Faculdade de Economia da Universidade do Porto. A sua área de investigação é a aprendizagem automática, principalmente em fluxos contínuos de dados. Publicou mais de 120 artigos científicos em conferências e revistas internacionais. Organizou várias conferências internacionais e séries de *workshops* em *Data Streams Analysis*. É membro do *Editorial board* de várias revistas internacionais nas áreas de aprendizagem automática e extração de conhecimento em bases de dados. É autor de um livro recente em *Knowledge Discovery from Data Streams*.

André C. Ponce de Leon Ferreira de Carvalho, Professor Titular da Universidade de São Paulo, possui Graduação e Mestrado em Ciência da Computação pela Universidade Federal de Pernambuco (1987), PhD em *Electronics* pela University of Kent (1994), Reino Unido. Os temas de investigação focam na área de aprendizagem automática, principalmente em computação bioinspirada e sistemas inteligentes híbridos, tendo orientado vários alunos de mestrado e doutoramento. Possui vários trabalhos publicados em diversos periódicos e conferências, além de ter organizado conferências e números especiais de periódicos. É membro do CEPID – Centro de Matemática e Estatística Aplicada à Indústria do ICMC-USP e coordena o Núcleo de Aprendizado de Máquina em Análise de Dados da Universidade de São Paulo.

Katti Faceli é Professora Adjunta da Universidade Federal de São Carlos, Campus de Sorocaba. É doutorada em Ciências da Computação e Matemática Computacional pelo ICMC-USP-São. Publicou diversos trabalhos em periódicos e conferências internacionais nos temas de bioinformática, análise de agrupamento e sistemas inteligentes híbridos.

Ana Carolina Lorena é Professora Adjunta do Instituto de Ciência e Tecnologia (ICT) da Universidade Federal de São Paulo (UNIFESP), em São José dos Campos-SP, Brasil. É doutorada em Ciências da Computação e Matemática Computacional pelo ICMC-USP-São Carlos. Publicou trabalhos em periódicos e conferências internacionais nos temas de classificação de dados e sistemas inteligentes híbridos.

Márcia Oliveira é estudante de Doutoramento na Faculdade de Economia da Universidade do Porto e investigadora no LIAAD-INESC TEC, o laboratório de Inteligência Artificial e Análise de Dados da Universidade do Porto. É especialista em análise de redes sociais.

Esta obra apresenta os temas clássicos e as tendências atuais nas áreas de aprendizagem automática, reconhecimento de padrões e análise de dados. Oferece uma perspectiva abrangente dos principais aspetos destas áreas. O conteúdo está organizado em três grandes tópicos: Análise Exploratória de Dados, Métodos Preditivos e Tópicos Avançados. O livro é orientado para estudantes de mestrado e doutoramento, introduzindo o leitor nos principais conceitos e algoritmos de aprendizagem automática e apontando caminhos para a sua implementação prática. Com uma abordagem equilibrada entre tópicos básicos e avançados e com um forte caráter didático o livro preenche uma lacuna de obras abrangentes e atualizadas voltadas para o público de língua portuguesa.

Em *Extração de Conhecimento de Dados*, os autores combinam as suas experiências no ensino e na investigação para apresentar os principais conceitos bem como a sua utilização em problemas reais. Este livro pode ser adotado como livro-texto ou material de apoio para estudantes de mestrado e doutoramento nas áreas de inteligência artificial, aprendizagem automática, análise de dados e sistemas inteligentes. Os leitores terão acesso a uma obra abrangente sobre os principais temas numa das áreas da informática e ciências da computação com maior crescimento e impacto industrial nos últimos anos.

Esta obra teve o patrocínio:



439

