Capítulo 1

Conceitos de Mineração de Dados Multimídia

Rafael Santos

Resumo

Avanços recentes em várias áreas tecnológicas possibilitaram um crescimento explosivo na capacidade de gerar, coletar e armazenar dados. O barateamento e popularização de dispositivos de coleta e reprodução multimídia e a Internet colaboram para que exista uma quantidade vastíssima de dados prontos para uso, mas em muitos casos sem uma estruturação que facilite a busca de informações interessantes ou relevantes.

*Mineração de Dados (*Data Mining*) é o nome dado a um conjunto de técnicas e procedimentos que tenta extrair informações de nível semântico mais alto a partir de dados brutos, em outras palavras, permitindo a análise de grandes volumes de dados para extração de conhecimento. Este texto apresenta conceitos de mineração de dados multimídia, (imagens, sons, vídeos e outros), algumas aplicações existentes e possíveis áreas de aplicação e pesquisa.*

1.1. Introdução

Estamos nos afogando em informação mas com sede de conhecimento – John Naisbitt, Megatendências.

Avanços recentes em várias áreas tecnológicas possibilitaram um crescimento explosivo na capacidade de gerar, coletar, armazenar e transmitir dados digitais. Na primeira década do século 21 já temos a possibilidade de armazenar vários gigabytes em dispositivos portáteis e alguns poucos terabytes em computadores pessoais a um custo acessível.

Sistemas interligados em rede permitem a coleta de dados de terminais simples, que são armazenados em grandes bases de dados centralizadas. Câmeras e filmadoras digitais permitem a captura de dados multimídia em vastas escalas a custo baixíssimo, programas de rádio e televisão podem ser armazenados digitalmente de forma relativamente simples, e a própria Internet é uma fonte praticamente inesgotável de dados multimídia que são coletados e armazenados de forma distribuída.

Dados coletados e armazenados podem ser de diversas naturezas e servir a diversas finalidades. Alguns exemplos de esforços de coleta de dados envolvendo grandes volumes são apresentados a seguir:[1]

- O LHC (*Large Hadron Collider*) é um acelerador de partículas instalado próximo da fronteira entre Suíça e França. Ele contém quatro detetores de partículas que registram 40 milhões de eventos por segundo, registrados por 150 milhões de sensores. O volume de dados pré-processados é aproximadamente igual a 27 terabytes por dia[2].

- O Instituto Nacional de Pesquisas Espaciais tem uma base de dados de imagens de satélite com mais de 130 terabytes [29].

- O projeto *Internet Archive*[3] mantém um arquivo de diversos tipos de mídia, contendo 2 petabytes e crescendo cerca de 20 terabytes por mês, com aproximadamente 130.000 vídeos, 330.000 arquivos de áudio, quase 500.000 documentos de texto e indexando 85 bilhões de páginas em várias versões.

- De acordo com algumas estimativas[4], o site YouTube continha 45 terabytes de vídeos em 2006. O site Flickr tinha 2 bilhões de fotografias digitais[5] em 2007 (e um teste rápido mostrou que já são ao menos 2.2 bilhões). Considerando que uma imagem, suas variantes criadas pelo site e outros dados como comentários ocupem um mínimo de 300 kilobytes, toda a coleção usa mais de 614 terabytes no total.

- O banco de dados GenBank contém coleções anotadas de sequências de nucleotídeos e proteínas de mais de 100.000 organismos, em um total de 360 gigabytes[6].

- O *Large Synoptic Survey Telescope* contém uma câmera digital de aproximadamente 3.2 gigapixels e deve coletar 20 a 30 terabytes de imagens por noite[7]. O projeto Pan-STARRS, quando completo, usará quatro telescópios, cada um com uma câmera de 1.4 gigapixels, para coletar aproximadamente 4 petabytes de imagens por ano. Como o levantamento será refeito várias vezes, poderá criar um "filme" de 10 terapixels em cinco bandas do espectro com 50 cenas, para detectar mudanças no espaço visível[8].

---

[1] Algumas destas estatísticas foram obtidas de sítios oficiais e algumas de fontes não confirmáveis como *blogs*. Não existe maneira de obter algumas informações sobre volume de bancos de dados de alguns serviços como YouTube, Google, etc. – para uma estimativa mais atualizada sugiro fazer novas buscas em sites especializados.
Algumas empresas como IDC (http://www.idc.com/) fornecem relatórios com estatísticas e estimativas de uso regional e mundial de armazenamento e uso de banda de rede, a custos bastante elevados.

[2] http://gridcafe.web.cern.ch/gridcafe/animations/LHCdata/LHCdata.html

[3] http://www.archive.org/index.php

[4] http://www.businessintelligencelowdown.com/2007/02/top_10_largest_.html

[5] http://www.techcrunch.com/2007/11/13/2-billion-photos-on-flickr

[6] ftp://ftp.ncbi.nih.gov/genbank/gbrel.txt

[7] http://www.on.br/glimpse/presentations/C-Smith.ppt

[8] http://www.on.br/newastronomy/presentations/J-Tonry.ppt

- Um levantamento feito pela *Winter Corporation*[9] menciona algumas bases de dados de grande porte em uso (em 2005): Yahoo! (100 terabytes), AT&T (93 terabytes), Amazon (24 terabytes), Cingular (25 terabytes).

Para ter uma idéia aproximada do que representam terabytes e petabytes usando outras medidas para comparação, consideremos:

- Um disco de um terabyte custa aproximadamente 200 dólares. Um ano de armazenamento dos dados do LHC custa então quase dois milhões de dólares – sem considerar a necessidade de armazenamento redundante.
- Para transmitir um petabyte de dados em uma rede com velocidade de 100 Megabits/segundo seriam necessários 86 milhões de segundos, ou quase 2 anos e 9 meses.
- Um petabyte pode ser gravado em 223.000 DVDs (4.7 gigabytes por DVD), que colocados dois por capa formariam uma pilha de 1.12 quilômetros de altura. Se o tempo de criação de cada DVD for de 15 minutos e tivéssemos cem computadores para criar os DVDs, seriam necessários mais de 23 dias para completar a gravação.

Apesar destes números apresentados serem relacionados a grandes projetos comerciais e científicos, podemos também observar efeitos da "avalanche de dados" no dia-a-dia: qualquer usuário de computadores com uso moderado da Internet tende a armazenar imagens, mensagens, documentos, vídeos em seus computadores, que podem facilmente ocupar dezenas de gigabytes. O número tende a aumentar para usuários frequentes de máquinas fotográficas e filmadoras digitais e apreciadores de música. Frequentemente estes usuários tentam impor uma organização à sua coleção particular de dados, e esta organização tende a ser feita usando *metadados* – informações sobre o conteúdo dos arquivos obtida de alguma forma, geralmente através da associação automática de informações relativas aos dados (por exemplo, nome de uma música) ou da análise dos mesmos (por exemplo, com uma nota de preferência ou estilo da música).

A capacidade de poder extrair informações contidas nos próprios dados digitais (com isto aumentando a quantidade e qualidade dos metadados) é altamente desejável, e pode ser atingida parcialmente com técnicas de busca baseadas em conteúdo (*content-based retrieval*). Recentemente técnicas de mineração de dados tem sido usadas [83, 100, 118] para derivar novos conhecimentos, conceitos ou estruturas a partir de dados digitais, em especial, multimídia; mostrando-se promissoras para pesquisa e aplicação.

O objetivo deste capítulo é familiarizar o leitor com os conceitos gerais de mineração de dados (e com o processo mais genérico de descoberta de conhecimento em bancos de dados) e com técnicas de mineração de dados aplicáveis à dados multimídia como imagens, sons e documentos na *World Wide Web*. Vários exemplos de aplicações serão apresentados com referências para que o leitor possa obter mais detalhes.

Este capítulo está dividido nas seguintes seções: esta introdução mostra o problema da "avalanche de dados". A seção 1.2 apresenta os conceitos de mineração de dados e suas principais técnicas, de forma genérica, com uma breve descrição de algoritmos

[9] http://www.wintercorp.com/VLDB/2005_TopTen_Survey/TopTenWinners.pdf

clássicos representativos das principais técnicas. A seção 1.3 comenta sobre os diversos tipos de dados e suas formas de representação, contrastando dados tabulares simples (mais usados em técnicas de mineração de dados) com dados multimídia, e mostrando como pode ser possível converter de um tipo para o outro. A seção 1.4 comenta sobre exemplos reais de mineração de dados multimídia e de tarefas semelhantes, contendo referências para artigos para aprofundamento. A seção 1.5 indica software que pode ser usado para mineração de dados em geral e a seção 1.6 apresenta algumas conclusões e sugestões de pesquisa.

1.2. Definição e Técnicas de Mineração de Dados

1.2.1. Definição

Mineração de Dados (em inglês *Data Mining*) é uma das fases do processo chamado Descoberta de Conhecimento em Bancos de Dados (ou KDD, do inglês *Knowledge Discovery in Databases*). Este processo é frequentemente confundido com mineração de dados em si, mas envolve outros passos e técnicas igualmente interessantes para o contexto deste curso, portanto merecendo uma descrição mesmo que simplificada.

O processo de descoberta de conhecimentos em bancos de dados é definido como **o processo não-trivial de identificação de padrões válidos, novos, potencialmente úteis e compreensíveis a partir de dados** (adaptado de [32]). O processo de descoberta de conhecimentos em bancos de dados é ilustrado na Figura 1.1.

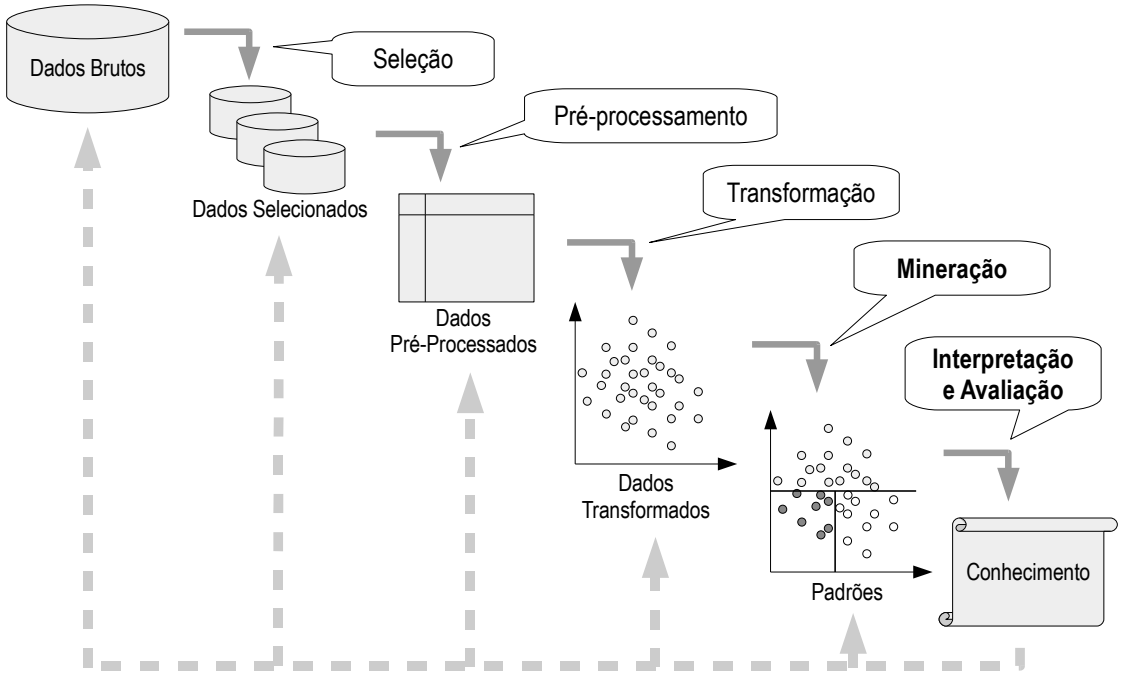


Figura 1.1. Processo de Descoberta de Conhecimento em Bancos de Dados (adaptado de [32])

Ainda de acordo com [32], e usando a Figura 1.1 como referência, podemos enumerar os passos do processo de descoberta de conhecimentos com a lista a seguir. Os passos da lista correspondentes às etapas do processo mostrado na Figura 1.1 são desta-

cados em negrito.

1. Compreensão do domínio da aplicação, do conhecimento prévio relevante e dos objetivos do usuário final do processo;

2. Criação de um conjunto de dados para uso no processo de descoberta através da seleção dos dados e/ou atributos relevantes (**seleção**);

3. Limpeza e pré-processamento dos dados, remoção de ruído e desvios (se possível e apropriado), decisão de como proceder com dados com atributos incompletos, normalização ou indexação de sequências temporais, etc. (**pré-processamento**);

4. Redução e reprojeção dos dados em outro conjunto de coordenadas, se necessário. Isto pode ser feito através da seleção de atributos úteis ou relevantes para representar adequadamente os dados sem perda de precisão, sempre dependendo do objetivo a ser alcançado. Se possível e desejável, mudar a representação dos dados para que a mesma seja invariante a aspectos que são relevantes (ex. escala, orientação) (**transformação**);

5. Escolha da tarefa de mineração de dados, considerando o objetivo genérico do processo (classificação, regressão, agrupamento, etc., descritos na seção 1.2.2);

6. Escolha do(s) algoritmo(s) de mineração de dados, baseado no objetivo geral e na consequente estrutura imposta aos dados. A decisão do(s) algoritmo(s) envolve a escolha de modelos, parâmetros, formas de execução, etc;

7. Mineração dos dados em si: busca de padrões de interesse usando algoritmos e dados selecionados; (**mineração**)

8. Interpretação dos resultados da mineração de dados, inclusive avaliação dos padrões, regras, etc. encontrados pelo processo de mineração; (**interpretação e avaliação**)

9. Consolidação e avaliação dos conhecimentos obtidos, documentação e elaboração de relatórios, resolução de conflitos com conhecimentos previamente existentes.

Os passos do processo ilustrado pela Figura 1.1 não precisam necessariamente ser seguidos na ordem descrita: a descoberta de conhecimentos em bancos de dados é um processo iterativo e exploratório; portanto alguns de seus passos podem ser executados novamente dependendo do resultado de passos posteriores. É importante ressaltar também o papel da visualização no processo: técnicas de visualização podem ser usadas em vários passos do processo para a tomada de decisão sobre atributos, dados e algoritmos a ser usados. Este capítulo não cobre técnicas de visualização com detalhes, técnicas específicas aplicáveis a diversos tipos de dados podem ser encontradas nas seções correspondentes.

O passo do processo que nos interessa é justamente o da mineração de dados, embora seja imperativo dominar os passos intermediários pois estes influenciam diretamente no resultado do processo de mineração, em particular no caso de dados multimídia, como veremos nas outras seções deste capítulo.

Mineração de dados é o nome dado ao conjunto de técnicas que possibilita o aprendizado prático de padrões a partir de dados, possibilitando explicações sobre a natureza destes dados e previsões a partir dos padrões encontrados (adaptado de [114]). De acordo com [53], existem duas categorias principais de mineração de dados: *preditiva*, que envolve o uso de atributos do conjunto de dados para prever valores desconhecidos ou futuros para um conjunto de dados relacionado; e *descritiva*, que foca na descoberta de padrões que descrevem os dados e que podem ser interpretados por humanos. Ambas categorias podem envolver a criação de um modelo que descreve os dados e podem ser usadas para produzir mais informações sobre os dados analisados.

1.2.2. Técnicas de Mineração de Dados

Antes de descrever as técnicas de mineração de dados é necessário definir alguns termos. Esta definição fica mais clara se considerarmos que os dados a ser minerados estão representados em uma tabela normal ou planilha. Um **dado** (ou registro, ou instância) corresponde a uma linha desta tabela, e um **atributo** corresponde à uma coluna da tabela. Assumimos que todas as linhas devam ser consideradas para a mineração de dados mas os valores dos atributos de algumas podem estar faltando, e em alguns casos a tarefa de mineração envolve descobrir os valores inexistentes.

Assumimos também que os atributos podem ser de diferentes tipos: numérico, nominal (categorias), intervalar, textual, relacional, etc. – existem várias taxonomias para tipos de atributos [83, 86], mas que o mesmo atributo tem o mesmo tipo para todos os dados, isto é, se para uma determinada tarefa de mineração de dados tivermos um atributo "duração" do tipo numérico expresso em segundos, o mesmo atributo será usado para todos os dados (ou seja, na mesma base não teremos um dado com "duração" expresso em datas como texto). Mesmo se o atributo "duração" estiver faltando para um determinado dado, sabemos que o tipo é numérico e o valor deve ser dado em segundos.

k	A_1	A_2	A_3	A_4	A_5	classe
1	010	0.60	0.70	1.7	L	flor
2	060	0.70	0.60	1.3	L	flor
3	100	0.40	0.30	1.8	W	grama
4	120	0.20	0.10	1.3	P	árvore
5	130	0.45	0.32	1.9	W	grama
6	090	?	0.18	2.2	W	?
7	110	0.45	0.22	1.4	L	?

Tabela 1.1. Exemplo de dados para mineração em forma de tabela

Um exemplo de conjunto de dados representado em tabela, que ilustra estes conceitos, é mostrado na Tabela 1.1 (com dados e atributos fictícios sobre fotografias digitais). Nesta tabela temos sete registros, instâncias ou dados; e cada um tem seis atributos (A_1 a A_5 e classe). Os atributos A_1 a A_4 são numéricos, possivelmente em escalas diferentes. O atributo A_5 é discreto, representado por um caracter ('L', 'W' ou 'P'). A classe é discreta, podendo assumir os valores "flor", "grama" ou "árvore". Para alguns dados, o valor deste atributo não está disponível, sendo representado pelo símbolo '?'. Como exemplo, um

dos atributos numéricos também está faltando para o registro 6.

Um outro conceito importante é o de *espaço de atributos*. Podemos imaginar que cada dado em uma base (linhas na tabela mostrada como exemplo) é um ponto n-dimensional que pode ser até visualizado para duas ou três dimensões. Dados semelhantes devem aparecer geometricamente próximos no espaço de atributos, e a distância calculada neste espaço entre dois pontos é usada por várias técnicas de mineração de dados para representar semelhança e diferença entre os dados correspondentes. A ordem em que os dados aparecem na tabela é irrelevante para a distribuição destes pontos no espaço de atributos.

Com estas definições podemos descrever as várias técnicas usadas para criar os modelos usados em mineração de dados. Estas técnicas podem ser categorizadas nos seguintes tipos (de acordo com [53]):

- **Classificação**: Descoberta de uma função preditiva que consegue classificar um dado em uma de várias classes discretas que são predefinidas ou conhecidas. Um exemplo pode ser dado (usando a Tabela 1.1) seria a classificação do conteúdo de uma fotografia digital a partir de atributos medidos da imagem digital, no caso, determinação do valor do atributo "classe" para cada registro, a partir dos valores dos atributos A_1 a A_5.
  A função de classificação é criada usando-se os atributos de vários exemplos existentes de dados e de suas classes fornecidas de forma supervisionada. A classe deve ser um atributo de tipo discreto, e para que um bom modelo seja gerado, é necessário ter um conjunto razoável de dados completos para cada uma das classes consideradas para a tarefa.

- **Regressão**: Descoberta de uma função preditiva de forma similar à feita em classificação, mas com o objetivo de calcular um valor numérico real ao invés de obter uma classe discreta. Algoritmos de regressão podem ser usados para, por exemplo, atribuir uma nota numérica (como um fator de indicação) para um filme baseado em seus atributos.
  Assim como no caso da classificação, a função que calcula a nota poderá ser criada analisando exemplos de filmes, seus atributos e notas já existentes, onde a nota deve ser um atributo numérico.

- **Agrupamento ou *Clustering***: Descoberta de grupos naturais de dados que possivelmente indicam similaridade entre os mesmos. Dados agrupados em um mesmo grupo podem ser considerados parecidos o suficiente; e dados em grupos diferentes são considerados diferentes entre si. Diferentemente das técnicas de classificação e regressão, não existem classes ou valores predefinidos que podem ser usados para identificar as classes: os algoritmos de agrupamento formam os grupos considerados naturais de acordo com alguma métrica, para que possam ser processados posteriormente como objetos correspondendo à mesma categoria.
  A maioria dos algoritmos clássicos de agrupamento somente permite o uso de atributos numéricos, já que uma função de distância é usada para determinar a pertinência de um determinado dado à um grupo, mas extensões que consideram dados numéricos e não numéricos de forma separada podem ser criadas. Usando técnicas

tradicionais e os dados da Tabela 1.1 como exemplo, poderíamos descartar os atributos A_5 e classe (por não ser numéricos) e verificar se os dados podem ser agrupados em dois ou mais grupos naturais.

- **Sumarização**: Técnicas que permitem a identificação de uma descrição compacta e inteligível para os dados (ou para um subconjunto dos mesmos). Frequentemente é possível sumarizar os dados mesmo com alguma imprecisão, e o valor das técnicas é na capacidade de descrever os dados, não necessariamente em sua precisão. Uma sumarização grosseira pode ser feita com os dados da Tabela 1.1 e expressa com regras: imagens de flores tem o valor para o atributo A_1 menor do que 70; imagens de grama e árvore tem $A_1 > 70$; e a diferenciação entre imagens de árvores e de grama é dada pelos valores do atributo A_2: $A_2 < 30$ para árvore e $A_2 > 30$ para grama.

- **Modelagem de dependência**: Técnicas que permitem a identificação de um modelo que descreve dependências significativas entre valores de um atributo de um conjunto de dados ou parte dele ou entre valores existentes nos dados. Técnicas de busca de regras de associação (também conhecidas pelo nome genérico "carrinho de compras") podem ser consideradas técnicas de modelagem de dependência. As técnicas mais básicas de modelagem de dependência geralmente assumem que os tipos dos atributos usados são discretos ou discretizáveis no próprio algoritmo que implementa a técnica.

- **Detecção de mudança ou desvios (*outliers*)**: Técnicas que permitem a descoberta e identificação de dados que não se comportam de acordo com um modelo aceitável dos dados (ou, por exemplo, mudanças em séries temporais ou em dados indexados por tempo). Estas técnicas podem identificar mudanças ou padrões inesperados em todos os dados ou em um subconjunto.

Estas técnicas não são mutuamente exclusivas entre si: técnicas de classificação como árvores de decisão [87] ou regressão são muito usadas para sumarização, classificadores são usados para criar modelos para detecção de desvios, técnicas de modelagem de dependência podem ser usadas para determinar subconjuntos de dados para processamento especializado, e até mesmo técnicas híbridas que combinam aspectos de classificação e agrupamento podem ser usadas quando não for possível usar dados e categorias de forma confiável [97]. As técnicas mais usadas e os seus algoritmos mais conhecidos são descritos, de forma genérica, no restante desta seção.

Algumas das técnicas mais usadas para criação de modelos a partir de dados são as que envolvem o uso de funções para classificar dados em categorias discretas, e o ponto central das técnicas é justamente a criação da função. O processo geral de classificação é descrito na Figura 1.2.

Para criação de uma função de classificação é necessário ter uma coleção de dados que sejam representativos das classes em questão, ou seja, de dados que já tenham sido rotulados com as classes às quais pertencem. Estas classes devem ser atributos discretos. Com este conjunto faremos um *treinamento* que envolve a criação de uma função que saiba diferenciar ou associar os valores dos atributos destes dados às suas classes. Para

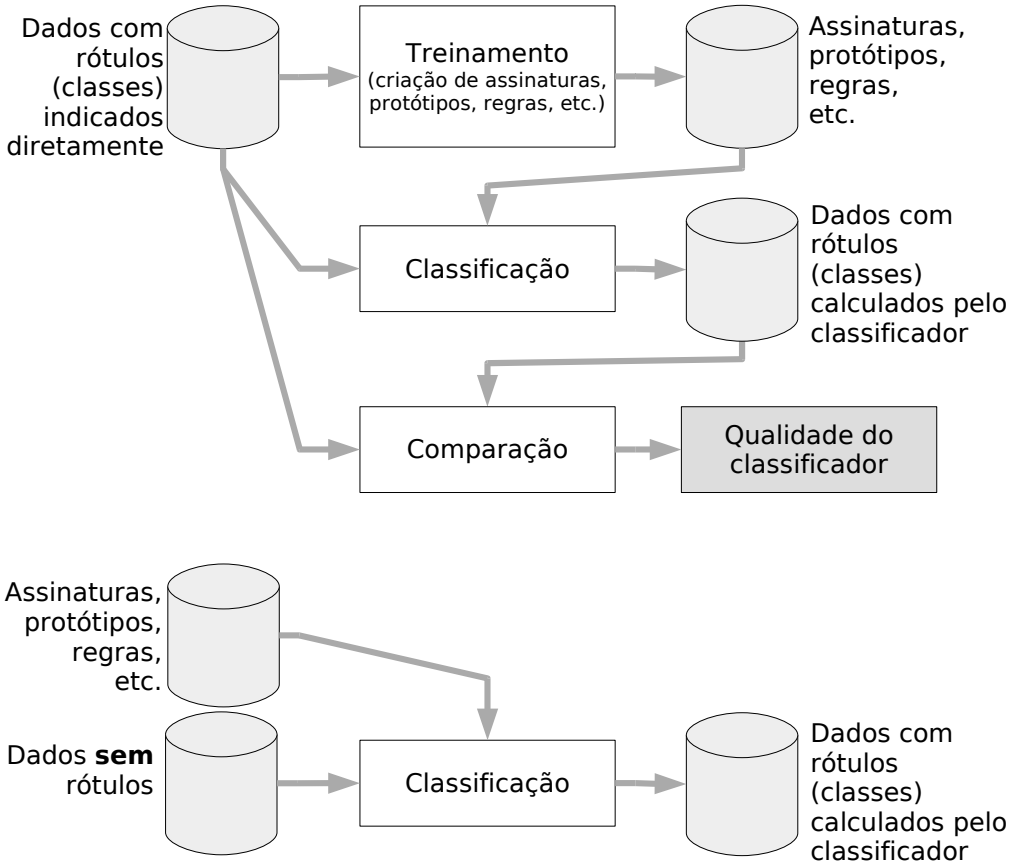


Figura 1.2. Processo de Classificação Supervisionada de Dados

isto transformamos os conjuntos de dados pertencentes à uma determinada classe em *descritores das classes*, que podem ser assinaturas, regras, protótipos, etc. daquela classe.

Podemos usar o conjunto de descritores e o algoritmo de classificação de duas formas: na primeira (mostrada na parte superior da Figura 1.2) classificamos os próprios dados usados para a criação do conjunto de descritores e verificamos se as classes obtidas são, como esperado, as mesmas das indicadas diretamente; com isto podemos avaliar a qualidade do algoritmo de classificação para aqueles dados. A segunda forma de uso é a mais comum (mostrada na parte inferior da Figura 1.2): usamos os descritores e o algoritmo de classificação para determinar o valor do atributo da classe para dados que não tenham este valor definido, efetuando assim a classificação em si.

Alguns dos algoritmos de classificação mais tradicionais e comuns são os que usam os valores dos atributos de forma combinada para delimitar regiões no espaço de atributos que definem as classes. Entre estes temos os algoritmos de árvores de decisão [87, 95] e método do paralelepípedo [89, 104]. Sistemas especialistas [17, 48], embora tradicionalmente construídos de forma supervisionada por *experts* no problema, podem ter suas regras criadas através da extração de informações sobre dados e classes, podendo ser considerados classificadores semelhantes às árvores de decisão. Estes três métodos também podem ser usados para sumarização pois é fácil obter regras que podem

ser compreendidas e avaliadas por usuários a partir das funções dos classificadores e dos descritores.

Redes neurais, em particular baseadas em perceptrons dispostos em múltiplas camadas [5, 31, 70, 104] e *Support Vector Machines* [20, 44] também podem ser consideradas métodos que particionam os dados usando os valores dos atributos: as partições separam os dados em diversas classes. A diferença fundamental dos outros algoritmos que particionam o espaço de atributos é que estes permitem a combinação de separações lineares mas não ortogonais aos eixos dos atributos, permitindo melhor precisão pelo classificador. A Figura 1.3 ilustra esta diferença.

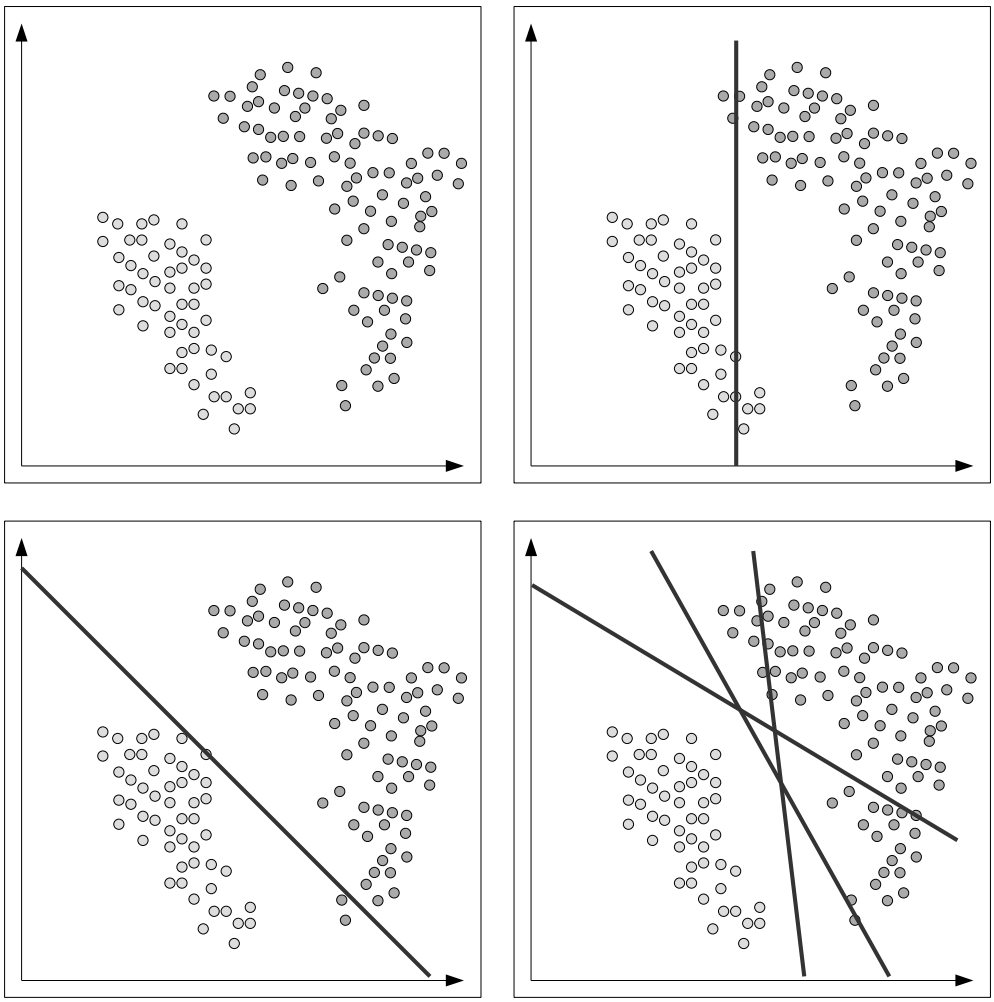


Figura 1.3. Separação de duas classes no espaço de atributos

Na Figura 1.3 ilustra, de forma bastante simplificada, o resultado da aplicação de classificadores que criam partições ortogonais aos eixos dos atributos (como sistemas especialistas básicos e árvores de decisão) e classificadores que criam partições não-ortogonais ou combinações (como redes neurais). Na parte superior esquerda da Figura 1.3 temos dados com dois atributos numéricos pertencentes a duas classes distintas representados no espaço de atributos. Na parte superior direita da figura temos uma classificação

dos dados que simplesmente verifica o valor do atributo correspondente ao eixo X, criando uma regra bem simples para classificar os dados de acordo com um valor limiar para X. Pode-se observar que esta classificação, embora simples, causa alguns erros de classificação nos próprios dados usados para determinar o limiar.

Na parte inferior esquerda da Figura 1.3 temos uma classificação feita por uma rede neural com um único neurônio. A classificação é mais precisa do que com o limiar ortogonal, mas por outro lado, sua explicação em termos naturais é mais complexa. Na parte inferior direita da Figura temos uma combinação de partições que separa perfeitamente as duas classes, mas cuja explicação em termos naturais seria ainda mais complexa.

Outros algoritmos de classificação usam métricas de distância a protótipos das classes: os mais conhecidos são os que usam a mínima distância a protótipo [70, 89, 104] ou máxima verossimilhança entre distribuições de classes [89, 104].

Técnicas de agrupamento diferem fundamentalmente das de classificação pois não usam informações sobre classes predefinidas – estas técnicas procuram, usando métricas definidas, formar grupos onde dados no mesmo grupo são semelhantes entre si e diferentes, de acordo com esta métrica, de dados de outros grupos.

Um dos algoritmos de agrupamento mais conhecidos, e que serve de base para inúmeros outros, é o algoritmo K-Médias [49, 70, 95]. Este algoritmo iterativo usa como entrada um valor K correspondente ao número de grupos que deve ser formado; uma métrica de cálculo de distância entre dois registros, algumas condições de parada das iterações e os dados em si. O algoritmo cria K centróides com valores inicialmente randômicos, e itera primeiro marcando cada registro como pertencente ao centróide mais próximo e depois recalculando os centróides dos grupos de acordo com os registros pertencentes (ou mais próximos) a estes. Durante a iteração uma métrica de qualidade de agrupamento é calculada (por exemplo, o erro quadrático total considerando os grupos formados até então), podendo ser usada como um critério de parada: pouca variação deste valor entre duas iterações indica que o algoritmo está convergindo e mais iterações não são necessárias.

O algoritmo K-Médias tenta identificar agrupamentos hiperesféricos no conjunto de dados, sendo adequado quando os dados tem uma distribuição desta forma, mesmo na presença de algum ruído; mas falhando quando a distribuição dos dados no espaço de atributos é irregular ou alongada. O algoritmo também precisa, a cada iteração, calcular as distâncias entre todos os dados e todos os centróides, podendo ser computacionalmente caro para um volume muito grande de dados.

A Figura 1.4 mostra seis passos da execução do algoritmo K-Médias com $K = 3$ em um conjunto artificial de dados com dois atributos numéricos com valores entre 0 e 1, onde existem três grupos concentrados de pontos com uma quantidade considerável de ruído (pontos fora dos três grupos concentrados).

Os seis passos da execução do algoritmo K-Médias mostrados na Figura 1.4 correspondem, respectivamente, à condição inicial (onde nenhuma iteração foi realizada, portanto os dados não são considerados pertencentes à nenhum dos grupos) e às iterações números 1, 2, 3, 10 e 20. A partir da primeira iteração os dados são marcados com tons de cinza diferentes, para facilitar a identificação dos grupos formados. Pode-se observar que os centróides dos grupos (indicados por pequenas cruzes) mudam sua posição, tentando

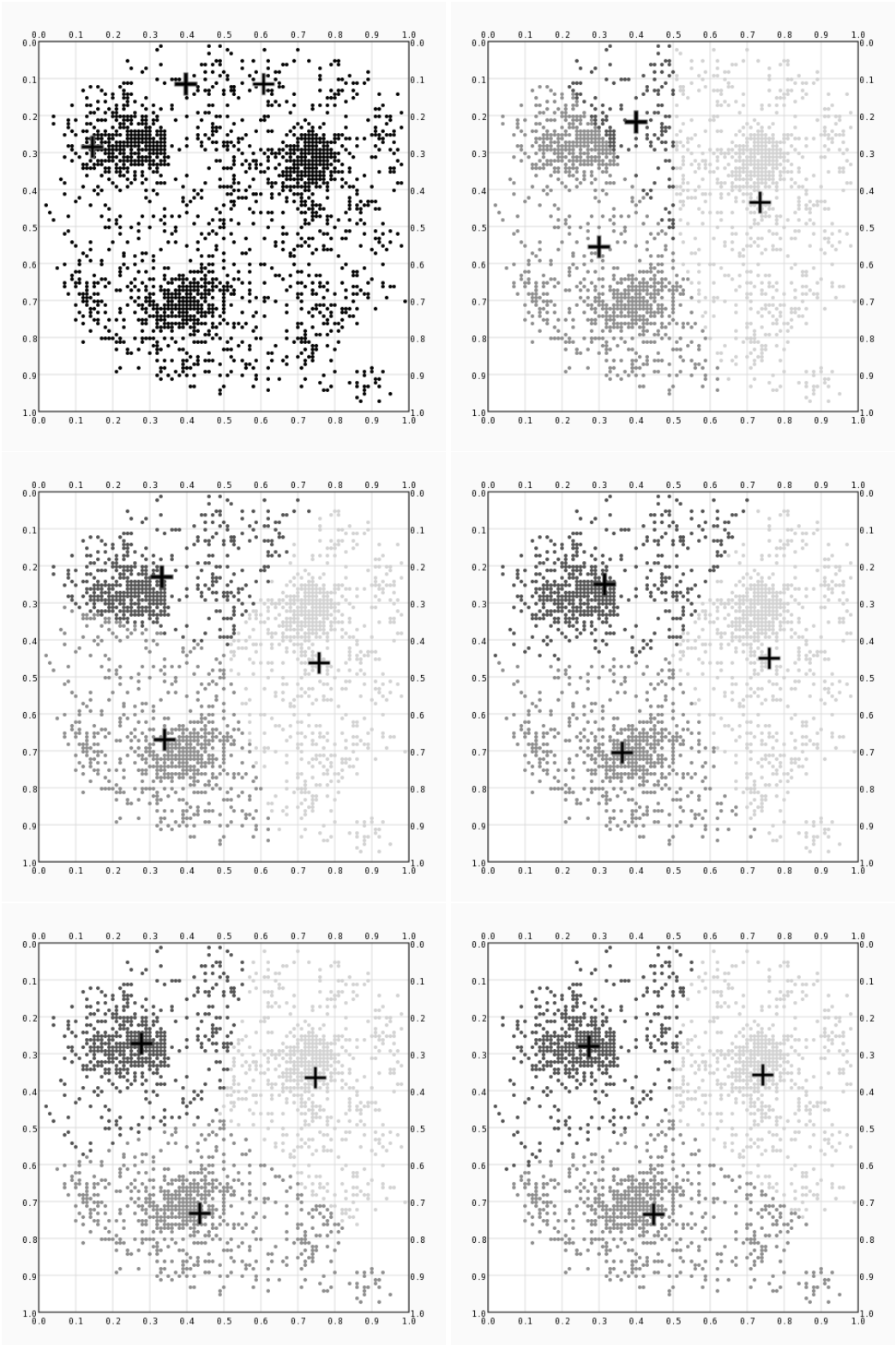


Figura 1.4. Passos do algoritmo K-Médias

se aproximar dos centros dos três grupos existentes. Nas primeiras iterações podemos observar claramente as mudanças das posições dos centróides, mas em iterações posteriores os mesmos quase não se movimentam, indicando que o algoritmo está convergindo.

Outros algoritmos usam conceitos semelhantes aos usados no K-Médias: o mais conhecido é o Fuzzy C-Médias [6], que usa conceitos de lógica nebulosa para calcular a pertinência de um dado a um grupo como sendo um valor contínuo entre 0 e 1 (enquanto o K-Médias considera pertinência booleana: um dado pertence a um grupo e a somente este grupo). O algoritmo Fuzzy C-Médias mantém, durante a sua execução, uma tabela de pertinência que indica o quanto cada dado pode ser considerado pertinente a cada grupo, esta tabela pode ser usada para verificar agrupamentos feitos de forma incorreta e para explorar outras possibilidades de pertinência [96, 97].

O algoritmo Fuzzy C-Médias [7, 22, 82, 98] tem muitas variantes, que permitem a criação de agrupamentos alongados (para dados com distribuições hiperelipsoidais) ou distribuídos regularmente nas bordas de hiperelipsóides (mas não nos centros). Este algoritmo também possibilita o cálculo de várias métricas de qualidade dos agrupamentos, facilitando assim a escolha do número de grupos C ideal dentro de um número de candidatos.

Um outro algoritmo, tradicionalmente usado para agrupamento de pixels de imagens de satélite, é o Isodata [70, 49]. Este algoritmo usa o K-Médias como base e algumas heurísticas de análise de agrupamento para evitar que o algoritmo forme grupos muito grandes ou muito pequenos. Este algoritmo consiste de vários passos e requer a definição de alguns parâmetros para uso pelas heurísticas.

Os Mapas Auto-Organizáveis de Kohonen [31, 58] são um tipo de rede neural que também pode ser usada como técnica de agrupamento (embora, a rigor, sejam técnicas de redução de dimensionalidade dos dados). Este algoritmo mapeia os dados originais em um novo espaço de atributos: uma matriz de neurônios de duas ou três dimensões que preservará a topologia dos dados originais (que frequentemente são representados com um número de dimensões mais elevado). Diferentemente de outros algoritmos de agrupamento, esta rede neural não fornece um número específico de grupos, mas os neurônios na matriz podem ser considerados representativos dos dados, e sua análise permite o uso como grupos não-exclusivos.

Outra categoria de algoritmos de agrupamento são os hierárquicos [4, 49], que usam um princípio diferente dos particionais (como K-Médias e Fuzzy C-Médias), que tentam de forma iterativa criar um número determinado de partições que definem os grupos de dados. Algoritmos hierárquicos criam partições juntando ou separando grupos sucessivamente de forma que é possível analisar todas as possíveis partições dos dados em grupos. Algoritmos hierárquicos *bottom-up* ou aglomerativos iniciam colocando cada dado da base em um grupo, e tentam sucessivamente juntar os dados/grupos mais próximos de acordo com uma métrica, até que todos os dados sejam unidos em um único grupo. Uma matriz de distâncias é usada durante a execução do algoritmo para determinar que dados/grupos devem ser unidos em cada passo. O resultado pode ser visualizado em um *dendograma* que permite, visualmente, estimar um número adequado de grupos para o conjunto de dados. A Figura 1.5 mostra um dendograma parcial resultante do agrupamento dos dados das flores Iris (um exemplo clássico de aprendizado por máquina).

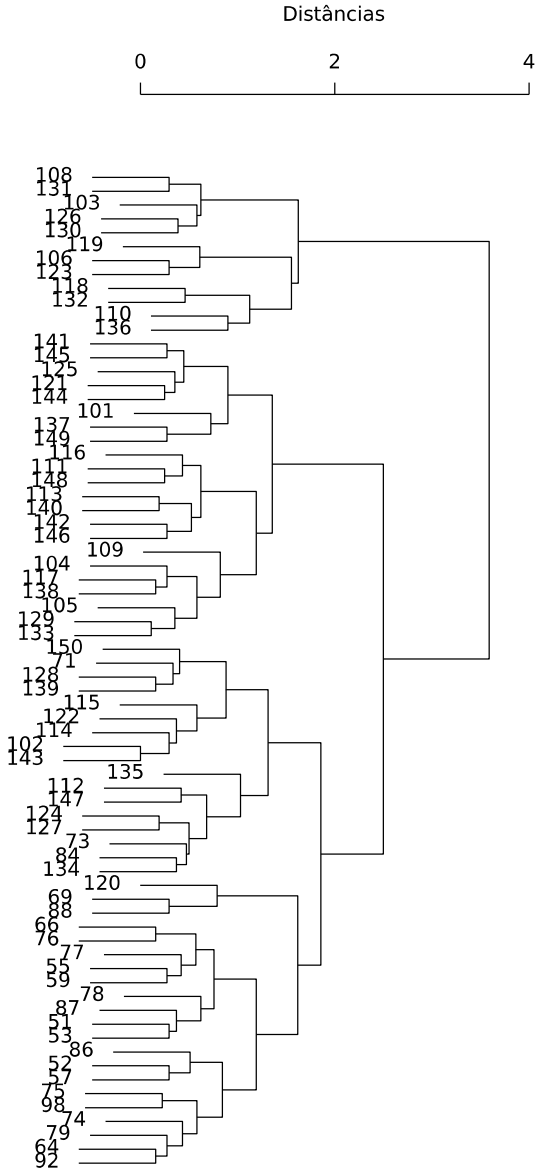


Figura 1.5. Dendograma parcial do agrupamento dos dados das flores Iris

Técnicas de sumarização permitem a descrição inteligível (por humanos) dos dados e de seu comportamento em geral. As técnicas mais usadas envolvem a criação de árvores de decisão [87, 95], que são um conjunto de testes sobre uma base de dados que indica a classe de cada dado a partir dos valores dos atributos de entrada. Os nós em uma árvore de decisão são testes sobre os valores dos atributos, e as folhas determinam as classes. A Figura 1.6 (adaptada de [62]) mostra um exemplo de árvore de decisão que descreve as decisões tomadas para classificar um possível cliente em relação ao risco de oferecer um empréstimo bancário, usando atributos como recursos e economias para a tomada de decisão.

Na Figura 1.6 os atributos usados para a decisão são indicados dentro de elipses, as decisões (classificações) dentro de retângulos e as arestas entre elipses e retângulos

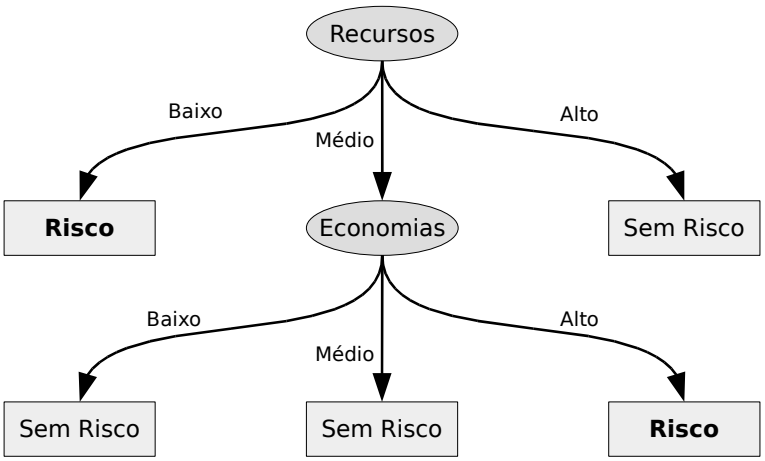


Figura 1.6. Árvore de decisão (adaptado de [62])

indicam os valores usados para as decisões.

Árvores de decisão são criadas por algoritmos que fazem o particionamento recursivo dos dados de uma base usando critérios como, por exemplo, entropia, tentando reunir em um dos galhos da árvore dados que pertençam, em sua maioria, à mesma classe. A vantagem principal das árvores de decisão é que elas explicam claramente que decisões são tomadas sobre quais atributos para classificação e sumarização; uma outra vantagem é que árvores de decisão podem ser *podadas* para obter uma sumarização mais compacta sobre os dados às custas de precisão de classificação. Exemplos completos de como montar árvores de decisão a partir de dados rotulados podem ser encontrados em [62, 96].

Técnicas de modelagem de dependência tentam identificar e descrever dependências significativas entre atributos ou valores de partes do conjunto de dados. Uma técnica bastante conhecida é a de identificação de associações ou co-ocorrências, que permite identificar, em um subconjunto de dados, os valores e atributos que ocorrem em conjunto com determinada frequência. O exemplo mais conhecido de aplicação destas técnicas é o chamado "carrinho de compras", cujo objetivo é descobrir, em uma lista de compras feitas em conjunto (no mesmo "carrinho"), quais objetos são comprados em conjunto.

O algoritmo mais conhecido de identificação de associações é o *a priori*. Este algoritmo tenta identificar regras do tipo *Se* **X** *ocorre então* **Y** *também ocorre*, onde **X** e **Y** podem ser itens em um carrinho de compras, ocorrências de valores discretos em registros, etc., podendo ser também combinações. Uma regra deste tipo (usando ainda o exemplo de carrinho de compras) seria *Se compra* **pão, manteiga** *então compra* **leite**.

Regras de associação devem ter métricas que indicam a significância e relevância das mesmas. Duas destas métricas são suporte e confiança. Usando ainda regras do tipo *Se* **X** *ocorre então* **Y** *também ocorre*, o suporte da regra é calculado como o número de eventos ou casos onde **X** e **Y** aparecem, dividido pelo número total de casos da base de dados. O suporte indica o quanto a regra de associação é significativa em relação à base de dados. A métrica confiança é calculada como o número de eventos ou casos onde **X** e **Y** aparecem, dividido pelo número de eventos onde **X** aparece, e indica o quanto **Y** é

relacionado com **X**.

Regras de associação podem ser usadas não somente com os problemas similares ao "carrinho de compras" mas também para identificação de fenômenos temporais. Regiões em séries temporais podem ser isoladas e tratadas como intervalos independentes e eventos podem ser indicados para que sua co-ocorrência seja analisada.

A implementação e uso do algoritmo *a priori* requer preparo especial dos dados para que possam ser representados adequadamente. Informações sobre pré-processamento, passos e exemplos de execução deste algoritmo podem ser vistos em [62, 96].

Outras técnicas de modelagem de dependência usam algoritmos mais complexos ou mesmo combinações de algoritmos para detectar dependências significativas entre subconjuntos de dados, identificando assim modelos que regem o comportamento de subconjuntos dos dados. Como estas técnicas são computacionalmente complexas, raramente podem ser usadas para tentar identificar modelos para um conjunto de dados largo.

Técnicas de deteção de mudança ou desvios (*outliers*) são comumente implementadas como um passo de algoritmos de classificação ou agrupamento: aplica-se um modelo que descreve os dados (criado por classificação ou agrupamento) a um conjunto de dados e usa-se uma métrica para avaliar a qualidade da classificação ou pertinência a agrupamento. Dados com baixa qualidade são candidatos a *outliers*. O problema com esta abordagem simplista é que deve-se tomar cuidado na construção do modelo, para que se possa ter certeza de que os dados identificados como *outliers* não são, por exemplo, correspondentes a uma classe nova ou que contenham valores extremos de atributos de uma classe existente. Este problema é ilustrado na Figura 1.7.

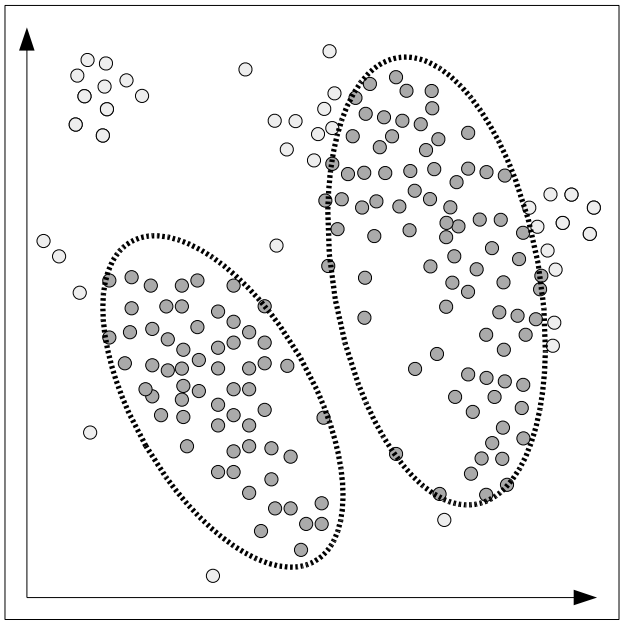


Figura 1.7. Possíveis *outliers*

Na Figura 1.7 (que mostra um conjunto de dados artificial com dois atributos numéricos) temos duas classes representadas por elipses (um algoritmo de classificação

supervisionada como o de máxima verossimilhança gera regras de classificação desta forma). Apesar das duas classes representarem adequadamente as distribuições dos dados, temos vários dados que estão fora da área delimitada pelas elipses. Em um caso (dados próximos à elipse maior) os pontos poderiam estar dentro do limite da elipse se a mesma fosse ampliada, e em outro caso (no canto superior esquerdo) é possível que os dados sejam representativos de uma classe até então desconhecida ou ignorada, e que não devam ser tratados como *outliers*.

1.3. Atributos e Pré-processamento

A maioria dos algoritmos de mineração de dados trabalha diretamente com dados tabulares ou relacionais, isto é, com dados estruturados de forma parecida com a mostrada na Tabela 1.1, onde cada linha corresponde a uma instância, registro ou dado; e cada coluna representa uma medida ou atributo sobre este dado. Dados tradicionalmente armazenados em bancos de dados, como, por exemplo, de transações comerciais, alguns tipos de *logs* de atividade em rede, alguns tipos de medidas científicas e outros são facilmente representados em tabelas, podendo ser usados diretamente pelos algoritmos. Vários dos algoritmos apresentados na seção anterior esperam que os dados sejam organizados em tabelas, preferencialmente com atributos numéricos.

Atributos de dados multimídia (imagens, áudio, vídeo, texto, web, etc.) devem ser extraídos e processados antes que possam ser minerados por algoritmos tradicionais. Seus atributos podem ser divididos em duas categorias: atributos obtidos ou extraídos diretamente do conteúdo dos dados (geralmente usando algoritmos específicos para extração de atributos de determinados tipos de dados) e atributos obtidos de outras formas, geralmente relacionados com o processo que gerou o dado multimídia. Este segundo tipo de atributo é chamado geralmente de *metadado*. A Tabela 1.2 mostra alguns exemplos de atributos que podem ser associados com dados multimídia, tanto do tipo extraído diretamente dos dados quanto metadados.

Os exemplos mostrados na Tabela 1.2 são genéricos e não cobrem toda a gama de atributos que pode ser extraída ou associada aos diversos tipos de dados. Por exemplo, existem diversas categorias de imagens digitais além das obtidas de câmeras: imagens digitalizadas de um *scanner*, por exemplo, não tem atributos como exposição, abertura, uso de *flash*, etc.; enquanto imagens de dispositivos médicos como Raios-X digitais podem ter atributos como orientação do aparelho, órgão a ser imageado, etc. que são necessários para este tipo de imagem mas que não tem sentido quando consideramos imagens de câmeras digitais. Ainda como exemplo, existem diversos tipos de texto não estruturado ou parcialmente estruturado (estórias, listas, entradas em enciclopédias, etc.), cada um destes tipos pode ter atributos que são relevantes para aquele tipo mas que não podem ser extraídos de arquivos de outros tipos.

É interessante observar que na coluna de atributos relacionados a conteúdo na Tabela 1.2 não existem atributos como objetos, seus nomes e características, como, por exemplo, nomes e atributos de pessoas ou objetos visíveis em fotografias digitais (que podem ser parte dos metadados se forem associados manualmente à fotografia) ou nomes de personagens principais em um livro. A extração e identificação destes atributos de mais alto nível semântico é complexa e sujeita a falhas, sendo frequentemente feita através de

Exemplo	Conteúdo	Metadado
Imagens de câmeras digitais	Cor predominante em regiões, histogramas, formas, cores e texturas de áreas perceptualmente homogêneas, disposição geométrica das áreas e relações espaciais entre elas, etc.	Dimensões da imagem, abertura, exposição, foco, data e hora da imagem, modo de operação, modelo da câmera, etc.
Arquivos de áudio digital (ex. música)	Ritmo ou tempo, timbre, pausas, informações derivadas de análise dos sinais de áudio, texto extraído da música através de reconhecimento de fala, etc.	Nome da música, autor, cantor, categoria ou estilo, duração, ano de produção, letra da música, etc.
Vídeo digital	Atributos das imagens estáticas que compõem o vídeo mais os obtidos da análise da diferença de imagens em sequência, atributos de áudio, correlação do conteúdo de áudio com o de vídeo, estimativa de movimento de objetos e da câmera, etc.	Contexto, duração, praticamente os mesmos atributos de imagens e de áudio, etc.
Texto em geral (não estruturado ou parcialmente estruturado)	Presença e ausência de palavras e associações, histogramas de palavras, comprimento de sentenças, métricas de complexidade do texto, etc.	Autor, língua, contexto, objetivo, tamanho do texto, palavras-chave, etc.
Documentos na WWW	Atributos de texto (similares aos usados para texto não estruturado), estrutura do documento, *links*, etc.	Dados sobre servidor, endereço, informações de *logs* de acesso, metadados do texto, etc.

Tabela 1.2. Alguns exemplos de atributos de dados multimídia

atenção humana, como será explicado ainda nesta seção.

Ainda sobre os exemplos da Tabela 1.2, podemos observar que alguns dos metadados só podem ser obtidos se tivermos acesso aos dados de uma forma organizada e planejada, frequentemente envolvendo a própria coleta dos dados – por exemplo, metadados sobre fotografias digitais são armazenados diretamente nos arquivos pelas próprias câmeras, mas podem ser perdidos em fotografias digitais armazenadas em *sites* ou durante a edição e retoque das fotografias, dependendo do formato usado para armazenamento. Como outro exemplo, dados sobre artistas, título da música, etc. não são armazenados diretamente nos arquivos em CDs de músicas, mas impressos no CD ou em um folheto que os acompanha; mas podem ser facilmente incluídos como *tags* se as músicas forem convertidas para outros formatos. Podemos esperar então alguma inconsistência nos metadados a não ser que possamos controlar todo o processo de coleta dos dados ou que possamos

confiar em que a fonte dos dados contenha os metadados organizados adequadamente.

Existem alguns padrões para descrição de metadados de forma unificada para facilitar a indexação e comparação [35], entre eles o *Dublin Core*, *RDF (Resource Description Framework)* e o *MPEG-7 (Multimedia Content Description Interface, ISO/IEC 15938)*. Este último é mais interessante pois permite a descrição de atributos de áudio e vídeo, mas é muito complexo e depende de implementação de algoritmos específicos.

Metadados podem não ser suficientes para descrever adequadamente o conteúdo dos dados multimídia, portanto a extração de atributos a partir do conteúdo é necessária. A automatização da extração é altamente desejável, mas existem três problemas que complicam esta automatização e que devem ser considerados para mineração de dados deste tipo:

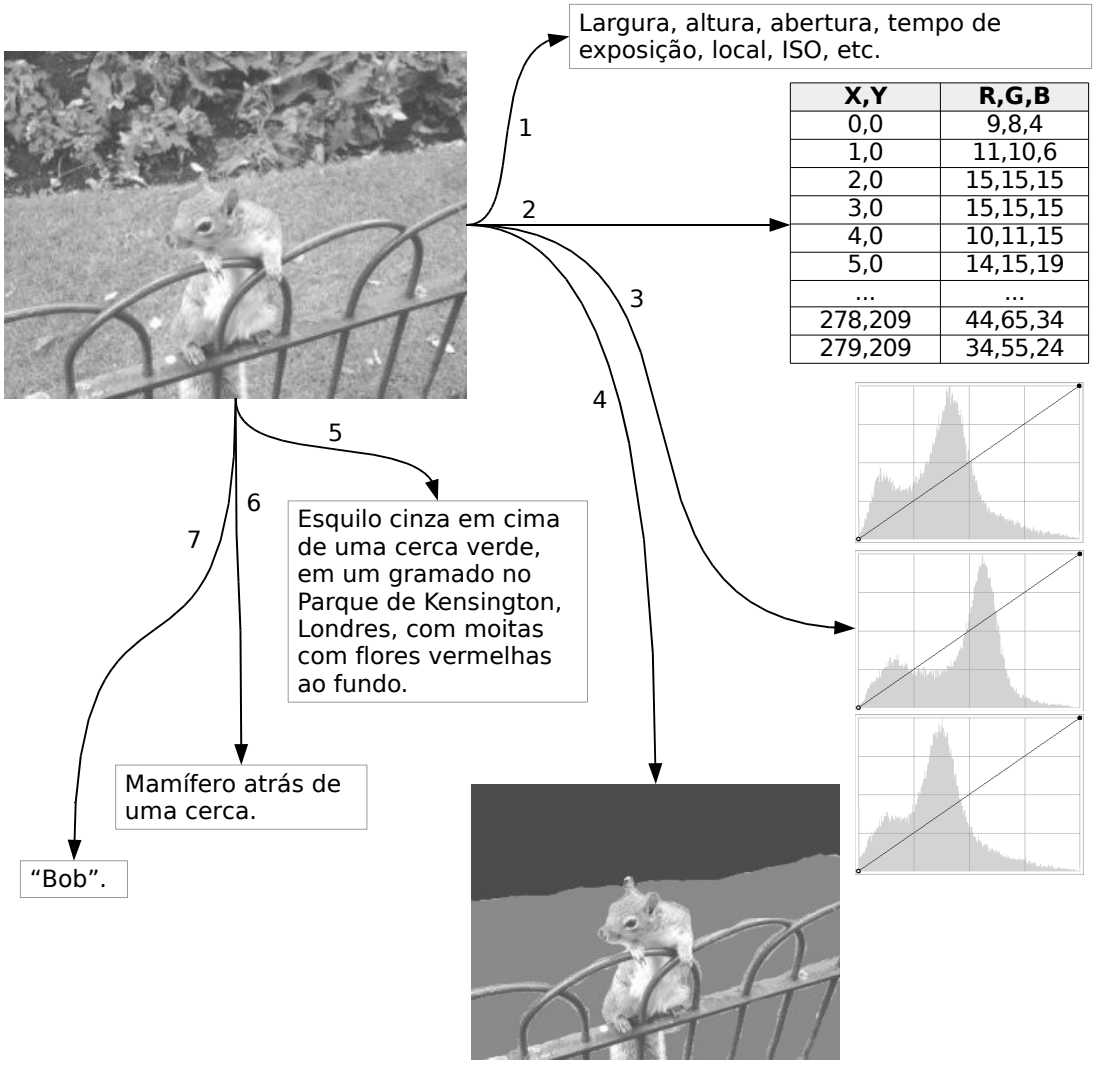
1. Para alguns tipos de dados a quantidade entre volume (por exemplo, em bytes, arquivos, etc.) e informações relevantes é muito desproporcional, exigindo grande poder computacional mesmo para a extração de atributos simples. Um exemplo clássico é o de vídeos digitais, onde várias cenas podem ser processadas para que sejam extraídas informações sobre objetos e seu comportamento, por exemplo – evidentemente este processamento pode ter que ser feito para muitos arquivos, aumentando o custo computacional.

2. Não existe uma forma simples, independente ou automática de definir quais atributos devem ser extraídos dos dados. Por exemplo, em uma tarefa de mineração de eventos significativos em vídeos, teremos passes, gols e faltas que são significativos em vídeos de jogos de futebol e ultrapassagens e batidas que são significativos em vídeos de corridas de automóveis. Dependendo da tarefa de mineração pode ser impossível extrair atributos genéricos para todos os dados existentes, sendo necessário trabalhar com um subconjunto de dados que seja bem restrito em relação ao objetivo desejado.

3. Existe uma diferença muito grande entre atributos perceptuais (isto é, que podem ser identificados e catalogados por seres humanos) e os dados em si, como representados em um computador. Em outras palavras, uma imagem, vídeo ou texto são para um computador, respectivamente, uma coleção de bytes em uma matriz, várias destas matrizes ordenadas no tempo (possivelmente com áudio sincronizado), e uma coleção de bytes em uma codificação qualquer; enquanto para um ser humano a imagem pode ser uma paisagem de praia, o vídeo pode ser um filme de ação e o texto pode ser *O Alienista* de Machado de Assis. Enquanto um ser humano pode facilmente identificar objetos semânticos em dados multimídia, computadores tem que usar várias técnicas, muitas delas complexas, e sem garantia de poder ao menos se aproximar da capacidade humana de identificação.
   A esta diferença entre os dados em sua forma mais básica e a informação contida naqueles dados dá-se o nome de *semantic gap*, ou distanciamento entre conteúdo e semântica. A resolução de problemas relacionados com o *semantic gap* ainda é objetivo de pesquisas avançadas, e o melhor que se pode esperar é a capacidade de identificar alguns objetos com semântica parcial nos dados.
   Deve-se comentar que mesmo seres humanos, capazes de extrair informações de

maneira fácil e rápida de dados multimídia, podem não concordar quanto aos atributos que podem existir nestes dados: por exemplo, uma pessoa pode indicar que existe em uma imagem uma flor, e outra pode, na mesma imagem, descrever uma rosa vermelha – embora as descrições pareçam iguais, sem um mapeamento semântico adequado elas podem ser completamente diferentes para algoritmos de mineração de dados.

Um exemplo destes problemas relacionados com a extração de atributos de dados multimídia é mostrado na Figura 1.8. Nesta figura temos uma fotografia de um esquilo (a imagem original é em cores) e vários tipos de atributos que podem ser extraídos desta imagem digital (alguns de forma automática). Os atributos são numerados na figura, e comentados a seguir.



X,Y	R,G,B
0,0	9,8,4
1,0	11,10,6
2,0	15,15,15
3,0	15,15,15
4,0	10,11,15
5,0	14,15,19
...	...
278,209	44,65,34
279,209	34,55,24

Figura 1.8. Problemas relacionados com extração de atributos de dados multimídia

Os dados e metadados que podem ser extraídos ou associados da imagem mostrada na Figura 1.8 são, respectivamente:

1. Metadados que podem ser obtidos da câmera digital usada. Estes metadados não precisam de atenção humana para ser associados à imagem, mas podem eventualmente ser perdidos com transformações e edições das imagens, e podem não ser relevantes para o processo de mineração.

2. Dados brutos da imagem (valores dos componentes vermelho, verde e azul de cada pixel). São a informação de nível semântico mais baixo deste dado multimídia, mas podem servir de base para extração de atributos de nível semântico mais alto, como regiões com formas, cores e texturas associadas, disposição e relações entre regiões, etc.

3. Histogramas que mostram a distribuição de níveis em cada um dos componentes da imagem. Também são de nível semântico baixo, mas podem ser usados por técnicas de análise de imagens para identificar atributos como cor predominante, brilho e contraste, etc.

4. Processamento parcial da imagem para separação de diversas categorias de frente e fundo (*foreground* e *background*) da imagem. Neste exemplo as regiões de frente e fundo foram extraídas manualmente, mas existem algoritmos que podem ser usados para extração de regiões homogêneas de imagens.

5. Descrição de alto conteúdo semântico feita por um interpretador humano. Esta descrição pode ser usada para estabelecer objetos (“esquilo”), relações (“em um gramado”), locais (“Kensington”) e outros, além de informações auxiliares que podem ou não ser úteis ao processo de mineração.

6. Outra descrição de alto conteúdo semântico, mas bastante simplificada quando comparada com a anterior, e que usa informação auxiliar para classificar o objeto já reconhecido da imagem.

7. Ainda outra descrição de alto conteúdo semântico, ainda mais simplificada, mas cujo significado e processo provavelmente não podem ser compartilhados com outros interpretadores – neste exemplo hipotético, um interpretador humano usou pistas visuais para identificar um esquilo em particular.

Deve-se observar que estes não são os únicos atributos que podem ser extraídos da imagem, muitos outros de baixo nível semântico podem ser extraídos dos pixels das imagens, e várias descrições de alto nível semântico diferentes entre si poderiam ser inferidas.

Para reforçar a importância de entender os problemas causados pelo *semantic gap*, vejamos as imagens mostradas na Figura 1.9. Destas imagens, as duas na fileira superior são do mesmo objeto, com diferença na escala, a terceira é de um objeto semelhante e a última de um aparentemente diferente. As imagens originais são em cores.

Descartemos os metadados das quatro imagens e consideremos somente o conteúdo para tentar responder uma questão simples: *estas imagens são semanticamente iguais?* Usando somente técnicas de processamento e análise de imagens para extração de regiões homogêneas e seus atributos (forma, cor predominante, textura), disposição

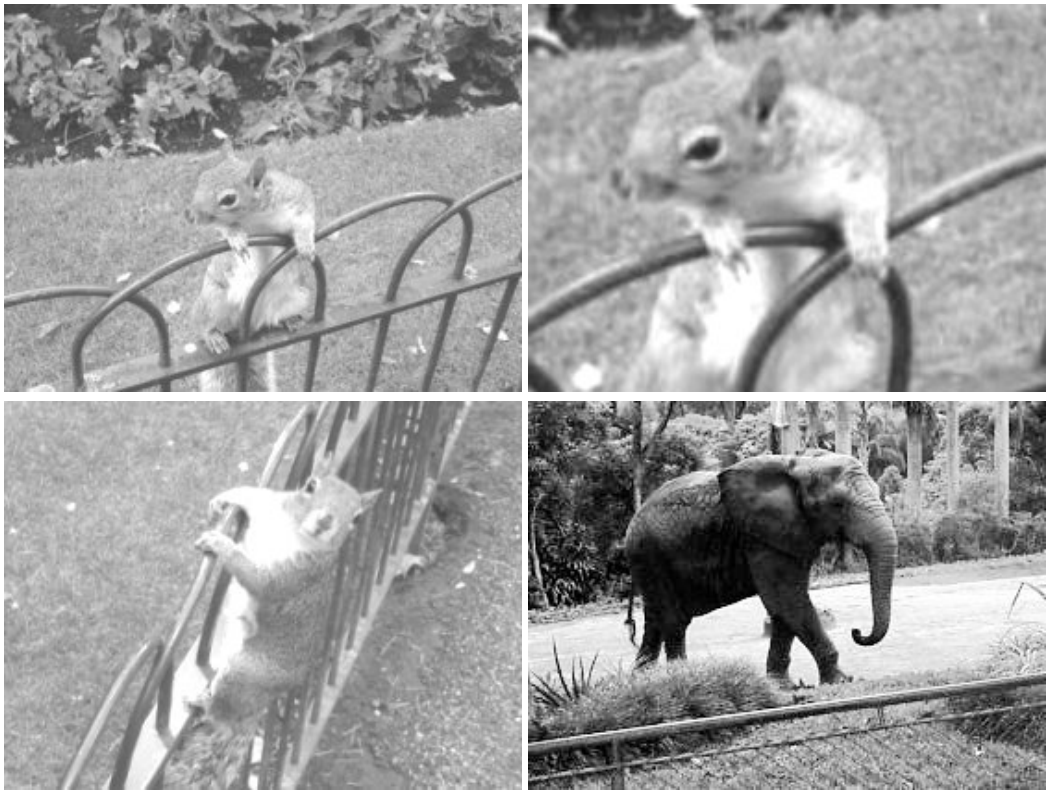


Figura 1.9. Objetos em dados multimídia que podem ou não ser considerados distintos

das regiões, etc. podemos chegar a descrições estatísticas sobre objetos perceptuais na imagem, separação entre objetos e fundo na imagem, etc., mas não podemos chegar ao ponto de transformar os pixels da imagem em descrições do tipo "esquilo em uma grade" de forma automática. O uso de contexto, supervisão e de experiência prévia é indispensável para a extração ou associação de atributos semânticos de alto nível.

Vamos assumir que existam descrições destas imagens feitas por interpretadores humanos, mas de forma livre (como feito nos exemplos mostrados na Figura 1.8), sem um guia sobre como fazer a descrição, isto é, sem um conjunto bastante restrito de atributos que possam ser usados por interpretadores. É possível, nestas condições, garantir a possibilidade de identificar uma imagem como sendo igual às outras?

A resposta para a pergunta não é definida, e não pode ser definida a não ser que tenhamos uma maneira de unificar ou simplificar as descrições feitas pelos interpretadores. As quatro imagens podem ser consideradas diferentes se descrições detalhadas sobre os objetos principais e secundários (fundo) forem feitas – a primeira seria diferente da segunda por ter mais informação ao fundo. As três primeiras imagens podem ser consideradas iguais por conter o mesmo objeto principal, "esquilo", se usarmos um conjunto mais restrito de atributos para identificação. As quatro imagens podem até mesmo ser consideradas semanticamente iguais se a descrição de seus objetos for bem genérica, como "mamífero" ou "animal em um parque".

Ainda outro exemplo de *semantic gap* pode ser dado por buscas em texto. Fer-

ramentas simples, existentes em praticamente qualquer sistema de informação, permite a busca por termos em textos, algumas permitindo o uso de expressões regulares para buscar variantes do texto. Ferramentas mais complexas podem permitir a busca por trechos de texto com combinações de palavras e/ou exclusão de termos, mas nenhuma destas capacidades realmente permite uma busca semântica, que pode ser algo como "liste nomes de políticos em uma edição de jornal" ou "procure trechos cômicos no texto" (conforme exemplo em [19]).

A extração ou associação de atributos corresponde aproximadamente aos passos de pré-processamento e seleção ilustrados na Figura 1.1, e é um dos aspectos mais complexos e críticos do processo de descoberta de conhecimento quando o objetivo é a mineração de dados multimídia. Infelizmente não existe (e nem podemos esperar) algoritmos mágicos que possam extrair atributos relevantes e robustos dos diversos tipos de dados multimídia, mas podemos usar atributos que podem ser extraídos automaticamente dos dados juntamente com (quando disponível) informações de nível semântico mais alto e mesmo associar informações de dados correlatos para enriquecer informações sobre o dado sendo considerado, para assim proceder à mineração. Estas técnicas são dependem bastante dos dados e do objetivo, e serão descritas através de exemplos na próxima seção.

1.4. Mineração de Dados Multimídia

Nesta seção veremos exemplos de aplicação de técnicas de mineração de dados multimídia para diversas categorias de dados e problemas. Dada a natureza complexa dos tipos de dados multimídia, muitas técnicas e aplicações apresentadas são relacionadas mais com suporte à mineração de dados (organização e representação dos dados, indexação, extração de atributos) ou pré-processamento do que com a mineração em si.

Esta seção é dividida em várias subseções, cada uma tratando de uma categoria de dados e exemplos de aplicações. Em alguns casos existe superposição entre as categorias, por exemplo, muitas técnicas relacionadas com séries temporais são aplicáveis a mais de um tipo de aplicação ou dado. Algumas subseções tratam de categorias de dados e aplicações que não são, estritamente falando, multimídia, mas compartilham do problema de mudança de representação dos dados e podem ser de interesse ao leitor. Onde possível referências bibliográficas sobre os problemas e soluções serão indicadas.

1.4.1. Aplicações: Dados Espaciais

Bancos de dados espaciais e Sistemas de Informações Geográficas (GIS, *Geographic Information Systems*) tem sido amplamente usados para diversas finalidades, em especial por órgãos e agências governamentais para mapeamento, planejamento, estatísticas e simulações, e mais recentemente por analistas de diversas áreas interessados em estudar o comportamento de dados em um contexto espacial. Existe um grande interesse em aplicar técnicas de mineração de dados a dados espaciais, preferencialmente integrando funções de mineração a sistemas existentes [73].

Dados espaciais neste tipo de aplicação são representados por objetos espaciais organizados em camadas temáticas como loteamentos, estradas, rios e lagos, etc. Os objetos espaciais podem ser pontuais, linhas, polilinhas, polígonos, etc., ou até mesmo células regulares ou imagens. Cada um destes objetos tem atributos espaciais (posição,

área, perímetro, forma, etc.) e atributos não-espaciais associados a ele. Bancos de dados espaciais e sistemas GIS provêem funções para armazenar, exibir, recuperar e gerenciar os objetos com seus atributos de forma unificada.

Técnicas de mineração de dados espaciais usam, tradicionalmente, os atributos espaciais juntamente com os não-espaciais e/ou extensões espaciais para algoritmos clássicos de agrupamento, descoberta de regras de associação e classificação. Outras diferenças entre mineração de dados espaciais difere de mineração de dados tradicional são [116]:

- Mineração de dados tradicional foca na descoberta de conhecimento global (sobre todos os dados); enquanto mineração de dados espaciais pode ser usada para a descoberta de informações espacialmente locais.
- Mineração de dados espaciais frequentemente usa o conceito de vizinhança espacial, pois espera-se que um fenômeno existente em um ponto do espaço tenha alguma correlação com fenômenos em posições espacialmente próximas.
- Mineração de dados tradicional usa alguns predicados de comparação em seus algoritmos (maior, menor, igual, etc.), enquanto mineração de dados espaciais pode usar vários outros predicados posicionais e relacionais como dentro, fora, perto, etc.

Como exemplos de técnicas de mineração de dados espaciais, temos métodos para extração de regras de associação que usam conjuntamente atributos espaciais e não espaciais para identificação de associações [60], uso de técnicas de mineração de dados espaciais para identificar padrões emergentes, que são associações onde o suporte é significativamente diferente entre classes, potencialmente indicando exceções à regras estabelecidas ou inferidas [13], identificação de *outliers* em dados espaciais para encontrar objetos próximos mas inconsistentes de outros [36], deteção de *clusters* coerentes no espaço e no tempo [52], etc.

Algumas técnicas tem sido desenvolvidas especificamente para tratar dados espaciais e espaço-temporais, com possibilidades de aplicação em mineração destes dados: histogramas espaço-temporais para otimização de buscas [30], estruturas de dados como árvores para indexação de grandes quantidades de dados pontuais [85] ou multidimensionais em geral [90], técnicas de integração de bases de dados espaciais disjuntas e dispersas [16], etc.

Várias aplicações tratam de problemas relacionados com mineração de trajetórias de objetos móveis sobre bases de dados geográficas, como coleta de trajetórias usando dispositivos móveis para mineração e identificação de trajetórias frequentes [76], monitoramento de movimento de objetos em espaços geográficos [71], descoberta de padrões de tráfego usando densidade de objetos móveis [66], indexação de bases de dados de trajetórias para otimização de buscas [37, 77], busca de objetos próximos de outros ao longo do espaço e do tempo [38], etc.

1.4.2. Aplicações: Dados Temporais

A maior parte dos dados coletados de forma automática tem alguma característica temporal, mesmo que implícita. Dados organizados como séries temporais podem ser usados

em aplicações que envolvem predição de valores (com algoritmos treinados para prever valores futuros da série a partir de eventos passados), localização de padrões repetidos ou característicos (*motifs*, [18, 67]) na série, criação de regras para descrever o comportamento dos dados em função do tempo, identificação de associações em segmentos temporais próximos ou distantes ou segmentação de dados com comportamento semelhante em intervalos de tempo diferentes.

A Figura 1.10 ilustra um exemplo de série temporal multivariada simples, contendo 12 variáveis numéricas reais indexadas ao longo do tempo (medidas diárias de cotação de várias moedas contra o dólar americano). Nesta série temos 12 medidas numéricas para cada unidade do tempo, mas a estrutura de uma base de dados temporal pode ser bem mais complexa: podemos ter dados esparsos (ao invés de coletados de maneira uniforme ao longo do tempo), dados com características diferentes ao longo do tempo (por exemplo, ocorrência ou não de eventos) e até mesmo estruturas complexas como segmentos de dados espaciais associados a um determinado instante no tempo.

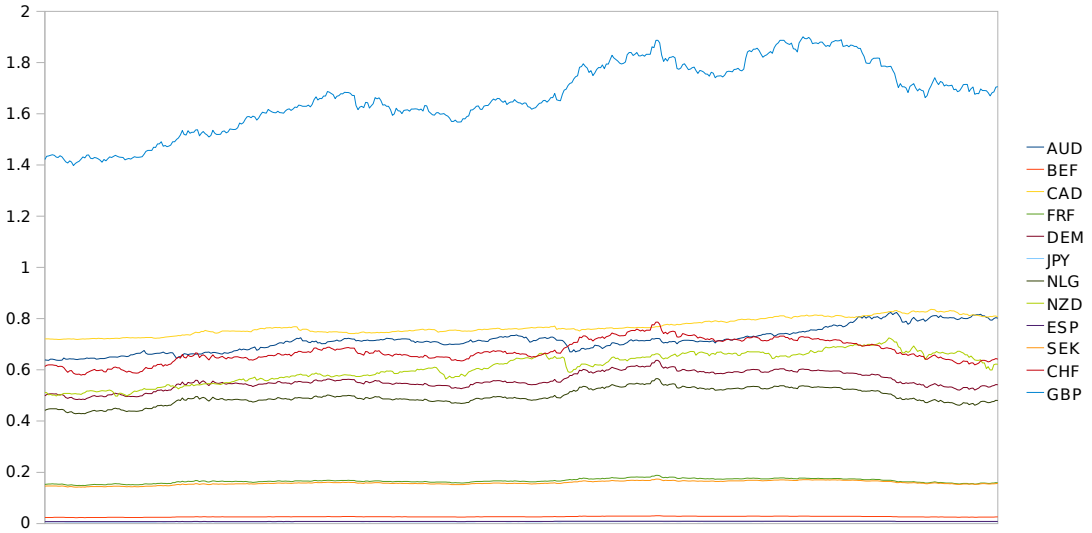


Figura 1.10. 500 dias de cotação de diversas moedas (obtidas no *site* http://www.stat.duke.edu/data-sets/mw/ts_data/all_exrates.html)

Uma das complexidades relacionadas com mineração de dados temporal é justamente relacionada com a modelagem do tempo. Medidas de tempo podem ser cíclicas em várias frequências: fenômenos meteorológicos, por exemplo, seguem padrões que podem ser cíclicos em relação ao dia, ano ou ciclos mais longos. Medidas relacionadas com atividades antrópicas podem variar de acordo com medidas naturais ou artificiais de tempo – por exemplo, consumo elétrico depende da época do ano por causa das estações e duração da luminosidade natural e em função de dias da semana. É imprescindível ter um bom conhecimento sobre os dados para modelar as séries temporais adequadamente para que os algoritmos de mineração possam ser aplicados.

Parte do esforço de pesquisa em mineração de bases temporais é relacionado com o desenvolvimento de técnicas para indexação, comparação e análise de séries temporais [63], como *segmented dynamic time warping* [57] e *segmentation-based symbolic*

representations [46], que tentam simplificar a representação de séries temporais de dados reais para facilitar a indexação e rápida recuperação em buscas. Outras técnicas são a identificação de padrões ou subsequências de eventos em séries temporais sem restrições de janelas no tempo (isto é, que pode ser aplicada em grandes séries temporais) [11], identificação de padrões temporais que podem variar de escala (duração e amplitude) na série temporal [56], identificação de padrões de comprimentos variáveis em séries temporais [12] e de eventos temporalmente extensos (isto é, com durações e relações de co-ocorrência entre eventos) [55] entre outras.

Outras aplicações de mineração de dados temporais são: descoberta de regras em bancos de dados médicos que detecta regras de associação atemporais, de efeito em tempos curtos e longos [105], descoberta de possíveis epidemias através da análise de *logs* de consultas a uma base de dados médica [42], caracterização de usuários baseado em padrões de digitação para autenticação [78] e outras.

Várias aplicações e técnicas para processamento, representação e mineração de dados espaciais usam também o tempo como atributo (como, por exemplo, indexação e mineração de trajetórias de objetos móveis), e foram descritas na seção anterior.

1.4.3. Aplicações: Imagens

Como mencionado na seção 1.3 (veja os exemplos das Figuras 1.8 e 1.9) é imprescindível extrair ou associar atributos semânticos de alto nível de imagens para a sua mineração. Existem três paradigmas principais de associação ou extração de atributos de imagens, usados no contexto de recuperação de imagens baseadas em conteúdo (CBIR, *Content-Based Image Retrieval*) [19, 91]:

- Anotação manual de imagens, para associar atributos textuais a cada imagem (podendo também ser aplicada a vídeo). As anotações podem ser na forma de texto (livres) ou de lista de presença ou ausência de objetos (direcionadas). Algoritmos de busca por conteúdo podem então ser usados para buscar por texto associado ao conteúdo; e algoritmos de mineração de dados podem ser usados para, por exemplo, encontrar imagens através de busca em regras de associação dos termos.
  Existem dois problemas associados a este tipo de paradigma: um é causado pela enorme quantidade de dados que devem ser observados e anotados, em especial quando consideramos que o trabalho deve ser feito por humanos; outro é relacionado com a subjetividade que pode causar anotações distintas para dados semelhantes (veja exemplo da Figura 1.9). O segundo problema pode ser resolvido parcialmente se usarmos múltiplos interpretadores por imagem (o que pode aumentar o efeito do primeiro problema) ou se ao invés de anotações livres forem usados formulários para anotações direcionadas (o que causa algumas limitações fazendo com que a técnica só seja efetiva para dados em um domínio preciso, como, por exemplo, radiologia ou histologia).

- Anotação automática de imagens, para associar às imagens atributos calculados a partir do conteúdo das mesmas. A grande dificuldade nesta abordagem é o *semantic gap* entre os atributos que podem ser facilmente extraídos de forma automática e os atributos de alto nível semântico que são de interesse [91].

- Um paradigma híbrido baseado em anotações dos dados multimídia. Este paradigma usa imagens com atributos semânticos de alto nível extraídos de forma supervisionada e tenta associar estes atributos a imagens que tenham regiões semelhantes.

Uma das formas de obter atributos de alta qualidade semântica de forma semi-automática é com o uso de técnicas híbridas baseadas em "sacos de atributos" (*bags of features*). Algumas imagens são processadas para a extração de seus atributos, usando regiões significativas (isto é, que tenham um tamanho acima de um limiar e que sejam homogêneas em relação à cor, brilho, textura, etc.). Atributos destas regiões são rotulados manualmente e associados à imagem, formando o "saco" de atributos, onde cada região é associada aos atributos da imagem (baixo nível) e aos identificados por humanos (alto nível). Outras imagens podem ser processadas para extração de regiões, e podemos inferir atributos semânticos de alto nível para estas regiões procurando regiões com atributos semelhantes no conjunto de imagens que foi rotulado manualmente.

Para um exemplo simplificado deste tipo de técnica, vejamos as imagens mostradas na Figura 1.11 (as imagens originais são coloridas). Usando técnicas de processamento de imagens podemos extrair algumas regiões significativas da primeira imagem (canto superior esquerdo) e rotular estas imagens, de forma supervisionada, como *céu*, *edifício*, *árvores* e *automóvel*. Cada região, além do rótulo semântico, teria medida estatísticas sobre textura, cor, etc. As outras imagens podem ser processadas da mesma forma automática gerando regiões que podem ser comparadas às regiões que já tem rótulos semânticos, e assim copiar estes rótulos para as suas próprias regiões.

Ainda usando as imagens na Figura 1.11 como exemplo, podemos rotular regiões como *edifício* e *árvores* nas outras imagens. Regiões que não sejam semelhantes às da primeira imagem (como *grama* e *estrada*, por exemplo) podem copiar os rótulos de outras imagens, não mostradas no exemplo, ou permanecer não-identificadas até que uma região semelhantes apareça na base de dados. Este processo pode ser parcialmente automatizado de várias formas, uma das quais é identificando regiões sem rótulos que aparecem em várias imagens e dando prioridade à identificação destas regiões.

Este tipo de técnica pode ser usado associado à *ontologias de objetos*, onde são mapeadas as relações entre diferentes formas de representar informação sobre objetos em imagens. Por exemplo, podemos mapear conceitos como *edifícios* e *casas* ao conceito genérico *construção* para que buscas e associações sejam mais flexíveis.

Técnicas semi-automáticas podem facilitar a tarefa de extração de atributos para grandes bases de dados de imagens, mas o problema de extrair atributos de imagens ou regiões com descritores semânticos ainda é bastante relevante. Técnicas de processamento de imagens podem ser usadas para extrair regiões através de segmentação da imagem [117] e associar atributos como forma, cor predominante, textura, etc. [21, 79] a estas regiões, mas sem associar informações semânticas.

Mesmo padrões modernos de descrição de dados multimídia como MPEG-7 [35] permitem somente a associação de metadados a cenas de vídeos em diversos níveis de detalhes e organização, mas sem extrair o conteúdo semântico para associação – é necessária a implementação de algoritmos que possam extrair as características da imagem que podem ser representadas por descritores de atributos [24]. Alguns destes descritores, rel-

Figura 1.11. Imagens com regiões semelhantes

evantes para imagens e *frames* de vídeo, são o descritor de *layout* de cores, o descritor de estrutura de cores na imagem, o descritor de textura homogênea e o descritor de histograma de bordas.

Técnicas de reconhecimento de padrões podem ser usadas juntamente com ontologias de objetos [88] em um esquema supervisionado para associar objetos conhecidos com atributos extraídos de imagens. Desta forma somente um conjunto reduzido dos dados teria que ser classificado de forma supervisionada, usando um classificador adequadamente treinado para identificar objetos em outras imagens e vídeos.

Diversos sistemas de recuperação de imagens baseados em conteúdo (CBIR) usam atributos das imagens para recuperação de outras imagens semelhantes a algumas mostradas ou a uma descrição ou *sketch* feito pelo usuário [19]. Os atributos usados são geralmente a cor predominante, cor de regiões, textura, forma, relações espaciais, etc. Outros sistemas usam mecanismos de busca interativa com eliminação supervisionada pelo usuário de imagens irrelevantes [69], técnicas de fusão de informação de textos e características de imagens como atributos de cor e textura [51] para melhor indexação e busca ou sistemas onde usuários criam filtros de forma colaborativa [10].

Algumas aplicações de mineração de dados em imagens são mineração de padrões de uso de solo em imagens de sensoriamento remoto usando modelos temporais [29], e o uso de aprendizado baseado em casos para identificar formas gerais de microorganismos e derivação de protótipos ou formas genéricas [84], classificação automática de imagens

de experimentos de cristalização de proteínas [112] e uso de técnicas de processamento de imagens, lógica nebulosa e modelagem espacial para associar informações semânticas a imagens de satélites de alta resolução para recuperação de regiões das imagens [41].

1.4.4. Aplicações: Áudio

Dados sonoros como música, trilhas de áudio obtidas de sequências de vídeo, fala, sinais que podem ser processados de forma similar a áudio (dados de sonares, outros tipos de sensores) são tão onipresentes e facilmente coletáveis quanto imagens, mas pouco explorados para tarefas de mineração de dados. A maioria dos esforços de pesquisa e desenvolvimento são na área de recuperação de informação musical (MIR, *Music Information Retrieval*) [80], a qual também tem como objetivo a indexação e descrição de música baseada em conteúdo.

Uma das dificuldades em descrever conteúdo de música é a quantidade de atributos que podem ser associados: timbre, tempo, tonalidade, ritmo, melodia, harmonia, estrutura, etc. [80]. Algumas destes atributos podem ser associados à parte da música, e alguns à musica como um todo. Existem formas unificadas de representação para alguns aspectos de música, como, por exemplo, partituras; mas o resultado final (que pode ser representado em um arquivo multimídia) é influenciado pelo estilo, ritmo, etc. do interepretador, podendo ainda ter ruídos e outras características que afetam o sinal armazenado. Conclui-se que é relativamente fácil partir de uma partitura para um arquivo com a música armazenado digitalmente, mas o reverso pode ser bem mais complicado.

Várias técnicas que podem ser usadas para indexação, descrição e mineração de séries temporais podem, potencialmente, ser usadas para sinais de áudio. Alguns exemplos são: combinação de classificadores baseados em máquinas de vetores de suporte (SVMs, *support vector machines*) [110], integração de métodos de agrupamento com modelos de mistura gaussianos para identificação de vozes [115] e mesmo linguagens para manipulação de arquivos de música integrais ou parciais em contexto de bancos de dados [109].

Algumas aplicações interessantes de mineração de áudio são: classificação de instrumentos clássicos em passagens solo usando técnicas espectrais e descritores de áudio MPEG-7 [101], separação de trilhas de áudio (obtidas de vídeos) em várias *meta-categorias* como fala, música, silêncio, som ambiente e combinações [121], técnicas de aprendizado ativo para identificação de emoção em fala [9], reconhecimento de música cantarolada (QBH, *query by humming*) [107], entre outras.

1.4.5. Aplicações: Vídeo

Muitos problemas, aplicações e soluções de mineração de vídeos são relacionados ao pré-processamento e extração de atributos de imagens e de áudio. Estes atributos são complementados com outros que são inerentes a vídeo, como diferenças temporais entre as *frames*, transições entre as *frames* e sincronismo de áudio e vídeo, entre outros.

Atributos para mineração de vídeo também podem ser derivados e indexados de várias fontes complementares, como, por exemplo, legendas (através de reconhecimento ótico de caracteres) [81] e presença de elementos esperados com algum significado (como, por exemplo, âncoras em noticiários [93]), etc.

Algumas aplicações de mineração de vídeo requerem a segmentação e indexação de trechos de vídeo ou conjuntos de *frames* e tentativa de associar significado a estes trechos. As técnicas são, como esperado, específicas para determinados domínios. Existem técnicas para acompanhar objetos específicos em vídeos, como acompanhamento da posição de jogadores e da bola em jogos de futebol [45, 103], detecção de logotipos em transmissões de televisão com possíveis aplicações em deteção de cópias não autorizadas e remoção através de *inpainting* [111], segmentação de imagens que contém atletas em vídeos de esportes para deteção e reconhecimento [65], segmentação de cenas de vídeos de noticiários usando deteção do âncora e segmentos de áudio [93] e segmentação de vídeos de esportes usando áudio [120].

Algumas técnicas com aplicações variadas são: localização e indexação de objetos que aparecem em cenas distintas [3] usando descritores invariantes de regiões, redeteção de objetos usando descritores de textura e cores [99], técnicas como histogramas de vídeos para deteção de similaridade e duplicação de segmentos mesmo com alterações nos vídeos [68], técnicas para deteção de cópias de vídeo baseadas em conteúdo (CBCD, *content-based copy detection*) [64] entre outras.

1.4.6. Aplicações: Texto

Existem estimativas de que a maior parte dos dados disponíveis em forma digital não é estruturada [33] – em particular, existe muita informação digital na forma de texto com pouca ou nenhuma formatação. Como a produção de documentos de texto é mais simples e imediata, sem a necessidade de equipamentos caros ou complexos, o volume (em número de documentos) e o número de fontes de origem e armazenamento é considerável, o que justifica o estudo e aplicação de técnicas de mineração de dados para obtenção de informações sobre estes textos.

Como nas outras tarefas que envolvem mineração de dados multimídia, é preciso primeiro extrair dos textos atributos que podem ser usados pelos diversos algoritmos. Apesar do pequeno volume em bytes de um documento, não é aconselhável usar todos os elementos do texto (ex. palavras) diretamente na mineração – regras de associação, por exemplo, podem encontrar redundância em nomes como “Albert” e “Einstein”, em diferentes contextos – podem ser textos sobre o cientista ou sobre o hospital.

A extração de atributos de texto pode ser feita de forma supervisionada por humanos ou automática. Processamento manual é inviável para grandes quantidades de documentos de texto, assim como para coleções muito dinâmicas (como mensagens em uma lista de discussão). Para fazer a extração automática de atributos devemos primeiro processar o texto para tokenização (separação do texto em elementos) e filtro de *tokens* irrelevantes, o que em si é um problema complexo e sem solução aparente porque não existe uma receita única que diga como palavras devem ser separadas para processamento adicional [113].

Consideremos, por exemplo, o símbolo ponto final, que pode ter diferentes significados em diferentes contextos: final de frase, marca de abreviação, separador decimal em algumas línguas, reticências, etc. O mesmo pode ocorrer com vírgulas e hífens, e o problema é ainda mais complexo para alguns domínios, como textos sobre química ou biologia, com nomes de fórmulas ou organismos compostos com os *tokens* ou textos sobre

matemática e física com equações formatadas em várias linhas de texto.

Outro passo possível no pré-processamento de textos para mineração é a deteção e separação do texto em unidades como sentenças e frases, o que pode ser complicado se a tokenização não for feita adequadamente (em geral o texto é separado em frases usando pontos, que, como mostrado, podem ter diferentes significados).

Ainda outro passo para o pré-processamento de textos é a normalização das palavras através de várias técnicas como extração de sinônimos (ex. carro→automóvel, residência→casa), redução para radicais, em especial de verbos (ex.: programando→programar, é→ser), etc.

Estes passos de pré-processamento ainda não transformam documentos de texto em descrições do seu conteúdo, o que é uma tarefa relacionada com processamento de linguagem natural (NLP, *Natural Language Processing*) [113]. Para aumentar as informações sobre a semântica do texto devemos aplicar outras técnicas como marcação de parte do discurso usando classes (adjetivos, verbos, pronomes, etc.), análise sintática, mapeamento em modelos e finalmente análise semântica. Todos estes passos são equivalentes, aproximadamente, aos passos de seleção, pré-processamento e transformação mostrados na Figura 1.1.

É importante observar que várias das técnicas de extração de atributos de documentos de texto são dependentes da linguagem, da categoria de texto (podemos usar regras mais específicas e diferentes entre si para mensagens de e-mail e artigos científicos sobre química, por exemplo) e de outros fatores que podem depender da tarefa, como, por exemplo, generalização de conceitos (devemos considerar "porco", "gado", "bife" e "carne" como conceitos diferentes)?

O uso de informações semânticas aumenta a complexidade do processo de mineração de textos e deve ser usado preferencialmente em domínios bem específicos [119]. Muitos algoritmos de mineração de texto evitam a tentativa de fazer uma análise sintática e semântica dos textos por causa da complexidade inerente e para evitar o atrelamento a um domínio ou categoria específica de problema, tentando extrair informações estatísticas e associações do texto de forma mais simples.

Uma forma bastante usada de fazer isto é através da técnica *bag-of-words* [34] que cria um vetor de atributos correspondente à ocorrência ou não de palavras em um texto. O texto pode ou não ser pré-processado para eliminação de termos inadequados, normalização, etc.; e o vetor pode ser binário (contendo valor verdadeiro para ocorrência de uma palavra e falso para não ocorrência) ou numérico, contendo um valor normalizado ou limitado de ocorrência de palavras. Os valores de pertinência de termos em documentos podem ter um fator de peso associado, que tenta fornecer uma medida de relevância do termo no documento. Algoritmos de mineração de dados podem usar então estes vetores numéricos para comparar documentos para classificação, agrupamento, busca de associações, etc. [27]

Algumas aplicações interessantes de mineração de textos (que também usam outras técnicas como mineração de grafos) são: agrupamento de artigos científicos usando referências bibliográficas [8], extração de termos que denotam sentimentos em textos não estruturados, com possíveis aplicações em monitoramento de opiniões em grupos de discussão na Internet e para pesquisa econômica ou de marketing [1, 74], análise de tendên-

cias temporais em documentos de texto sobre domínios específicos [47], entre outras.

Algoritmos de mineração de texto juntamente com algoritmos de mineração de grafos podem ser usados também em aplicações que processam informação textual altamente formatada ou organizada. Alguns exemplos são aplicação de algoritmos de agrupamento de documentos XML [61] para reduzir o número de documentos elegível para determinadas buscas, representação de documentos XML para mineração usando estrutura ou estrutura mais conteúdo [108], cálculo de métricas para medida de similaridade estrutural entre documentos XML [50], entre outras.

1.4.7. Aplicações: Páginas na WWW

A WWW (*World Wide Web*) é o maior e mais conhecido repositório de documentos hipertexto existente [14]. Documentos hipertexto diferem de documentos em texto pois além do conteúdo textual podem conter:

- Estruturas de metadados como parte do documento. Embora nem sempre presente, é possível associar um documento a um autor, URL, domínio, empresa, etc.
- Estruturas simples de formatação que, embora nem sempre presentes ou mesmo coerentes em um documento, podem servir para uma segmentação do documento em seções e subseções.
- *Links* ou elos com outros documentos, explicitados na própria estrutura do documento hipertexto, que podem ser usados como informações sobre associação, popularidade e/ou relevância.
- *Links* para objetos que devem ser associados ao próprio documento hipertexto, como referências a figuras e outros objetos multimídia. Estes *links* são usados para compor texto e objetos multimídia na mesma diagramação, pois não é possível inserir o conteúdo multimídia diretamente no hipertexto (isto é, o conteúdo multimídia não é inserido no conteúdo do documento). Para algumas aplicações pode ser importante distinguir entre *links* para objetos que estão no mesmo domínio e para objetos que estão em outros domínios.

Além destas características, arquivos hipertexto na WWW geralmente são associados a temas facilmente identificáveis: podemos assumir que documentos hipertexto obtidos em endereços do domínio `inpe.br` são relacionados à atividades desenvolvidas no instituto.

Documentos na WWW são controlados por servidores, que atendem a pedidos de clientes como navegadores e *crawlers* (ferramentas que varrem a Internet para indexação de documentos). Os servidores fornecem os documentos, que podem ser estáticos (armazenados como arquivos nos servidores) ou dinâmicos (criados a partir de parâmetros ou por sistemas dedicados). Os servidores armazenam em *logs* os pedidos de documentos (inclusive marca de tempo), o *status* da operação, os bytes enviados e algumas informações sobre o navegador. Estes *logs* também podem ser usados de diversas formas para mineração de dados, inclusive para medidas de associação sem *links* entre documentos e para identificação de sequências de navegação nos documentos do servidor.

Algumas tarefas de mineração de dados de hiperdocumentos são relacionadas com o pré-processamento para extração de atributos e enriquecimento dos mesmos com informações complementares, possivelmente obtidas de *logs* ou de documentos associados; ou com estruturação dos documentos e seus atributos para aplicação de técnicas especiais. Por exemplo, coleções de documentos na WWW podem ser representados como grafos, permitindo o uso de classificadores que consideram a estrutura interna do documento e não somente seu conteúdo [28]. Informações semânticas extraídas a partir dos textos também pode ser usas páginas também podem ser usadas para indexação e anotação, como, por exemplo, deteção e extração de termos geográficos e relacionais em páginas para indexação espacial [106] que aumenta a relevância com proximidade geográfica.

Outras técnicas tentam extrair regiões ou metadados de documentos para, por exemplo, identificar que regiões ou segmentos de um documento hipertexto contém conteúdo relevante ou potencialmente interessante, e que regiões e segmentos podem ser considerados ruído (propagandas, cabeçalhos, rodapés, etc.) [59]

Alguns objetivos em mineração de documentos de hipertexto é para indexação inteligente, usando, para determinar a relevância de um documento, não somente o conteúdo mas também outros atributos associáveis. Por exemplo, é possível adaptar e personalizar sistemas de busca usando classificação baseada em *feedback* não-intrusivo do usuário [25] ou usando informações adicionais como localização geográfica dos servidores e domínios das *URLs* [2].

Outras técnicas que tentam analisar conteúdo, histórico de acesso e outros dados para personalização de apresentação para usuários usam análise de *logs* de acesso a documentos na WWW para identificar preferências dos usuários em relação à categorias e objetos específicos [43], mineração de padrões significativos de uso (SUP, *Significant Usage Patterns*) para descobrir preferências e motivações de usuários de *sites* usando agrupamento de sequências de navegação e demonstrando o conceito com *logs* de um servidor de aplicações de um *site* de *e-commerce* [72], técnicas para manutenção de perfis de usuários de *sites* que se adaptam a mudanças no perfil de navegação [102], recomendação de outros documentos para usuários através da modelagem da necessidade do usuário (e não das correlações com atividades de outros usuários) [23], entre outros.

Documentos e coleções de documentos podem evoluir ao longo do tempo, e existem técnicas para analisar documentos e *hyperlinks* entre documentos considerando mudanças temporais para tentar identificar conjuntos de documentos que são recentes ou históricos [26].

Recentemente técnicas de mineração de dados tem sido aplicadas à mineração de grafos de redes sociais como Orkut, MySpace, Facebook, etc. Por exemplo, existem algoritmos para identificação de comunidades não-explícitas através da análise dos *links* dos documentos [44] ou que geram atributos que podem ser interessantes ao analisar nós de redes sociais ao invés de usar simplesmente os *links* entre os nós, demonstrando também uma aplicação em uma rede de citações bibliográficas [54].

Aplicações de mineração de conteúdos de *blogs* também tem sido alvo de interesse recente. Como exemplo de aplicação temos gerenciamento de reputação [39], feito através da análise de vários *blogs* e mapeamento de palavras em tabelas de conceitos.

Existem outras aplicações de mineração de *logs* (não só de servidores na WWW, mas também de outros tipos de servidores) com diversas finalidades: deteção de intrusão [40, 92, 75], identificação de vendedores potencialmente desonestos em serviços de leilão na Internet [15], etc.

1.5. Algumas Ferramentas

Uma forma de entender melhor os conceitos e imaginar aplicações de mineração de dados é pela experiência. Existe software que permite a exploração de técnicas de pré-processamento, diferentes algoritmos e formas de visualização em ambientes relativamente simples e usando computadores pessoais sem muitos requisitos de memória ou capacidade de processamento.

O software público de mineração de dados mais popular é o WEKA (*Waikato Environment for Knowledge Analysis*), desenvolvido na Universidade de Waikato, Nova Zelândia [114]. Este software é escrito em Java e contém implementações de vários algoritmos de classificação, agrupamento, previsão e busca de regras de associação, apresentados em uma interface gráfica que permite várias formas de interação. A API (*Application Programming Interface*) do Weka pode ser usada em aplicações independentes em Java sem necessidade de licenciamento comercial. O software e mais informações podem ser obtidos em `http://www.cs.waikato.ac.nz/ml/weka/`.

Uma outra alternativa interessante de software para exploração e aprendizado de mineração de dados é o RapidMiner (previamente conhecido como YALE (*Yet Another Learning Environment*). Neste software, experiências e testes de mineração são implementados como documentos XML, e podem ser executados através da interface gráfica, por linha de comando, em lotes ou integrados em outras aplicações em Java. Os operadores do software Weka também estão presentes no RapidMiner, que oferece ainda várias facilidades para mineração de textos, visualização multidimensional, operadores de pré-processamento e outros. A versão comunitária software, documentos e exemplos podem ser obtidos em `www.rapidminer.com`.

É importante observar que estas ferramentas podem não atender a uma necessidade específica, por não ter a implementação de um algoritmo específico ou por não poder processar os dados na forma em que são coletados (o que acontece frequentemente no caso de dados multimídia). Neste caso, é possível escrever módulos externos, independentes das ferramentas, para pré-processamento e conversão dos dados para uso pelas ferramentas, ou mesmo escrever aplicações que usam a API diretamente [94].

Algumas ferramentas comerciais de mineração de dados são Clementine, da SPSS e SAS Enterprise Mining da SAS. Algumas ferramentas são módulos associados a sistemas de gerenciamento de bancos de dados como IBM DB2 Intelligent Miner; Oracle Data Mining e Microsoft SQL Server Data Mining. Algumas suites científicas e matemáticas também tem módulos para mineração de dados: Statistica, da StatSoft e Matlab da MathWorks são alguns exemplos.

1.6. Conclusão

Foram apresentados a motivação, os conceitos básicos e alguns exemplos de aplicações de mineração de dados multimídia. A seção 1.4, que tratou das categorias específicas de mineração de dados multimídia mostrou alguns dos problemas associados à extração de atributos dos diversos tipos de dados, com referências para publicações que abordaram problemas específicos, significativos ou inovadores.

A maioria das publicações usadas como referências foram obtidas da série *Lecture Notes in Computer Science* da editora Springer, que publica vários anais de congressos na área, inclusive dos congressos *Advanced Data Mining and Applications – ADMA* , *Advances in Knowledge Discovery and Data Mining – PAKDD*, *Advances in Web Mining and Web Usage Analysis – WebKDD*, *Machine Learning and Data Mining in Pattern Recognition – MLDM*, *Principles of Data Mining and Knowledge Discovery Third European Conference – PKDD*, entre outros. Outras fontes de referências sobre o assunto podem ser obtidas através dos portais Google Scholar (`http://scholar.google.com`) e CiteSeer^X (`http://citeseerx.ist.psu.edu/`).

Referências

[1] Edoardo Airoldi, Xue Bai1, and Rema Padman. Markov Blankets and Meta-heuristics Search: Sentiment Extraction from Unstructured Texts. In *Advances in Web Mining and Web Usage Analysis: 6th International Workshop on Knowledge Discovery on the Web, WebKDD 2004, Seattle, WA, USA, August 22-25, 2004, Revised Selected Papers.*, pages 167–187, 2004.

[2] Mehmet S. Aktas, Mehmet A. Nacar, and Filippo Menczer. Using Hyperlink Features to Personalize Web Search. In *Advances in Web Mining and Web Usage Analysis: 6th International Workshop on Knowledge Discovery on the Web, WebKDD 2004, Seattle, WA, USA, August 22-25, 2004, Revised Selected Papers.*, pages 104–115, 2004.

[3] Arasanathan Anjulan and Nishan Canagarajah. Video object mining with local region tracking. In *Multimedia Content Analysis and Mining International Workshop, MCAM 2007, Weihai, China, June 30-July 1, 2007. Proceedings.*, pages 172–182, 2007.

[4] Eric Backer. *Computer-Assisted Reasoning in Cluster Analysis*. Prentice-Hall, 1995.

[5] R. Beale and T. Jackson. *Neural Computing: An Introduction*. MIT Press, 1990.

[6] James C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, 1st edition, 1987.

[7] James C. Bezdek and Sankar K. Pal. *Fuzzy Models for Pattern Recognition*. IEEE Press, 1st edition, 1992.

[8] Levent Bolelli, Seyda Ertekin, and C. Lee Giles. Clustering Scientific Literature Using Sparse Citation Graph Analysis. In *Knowledge Discovery in Databases:*

PKDD 2006 10th European Conference on Principles and Practice of Knowledge Discovery in Databases Berlin, Germany, September 18-22, 2006 Proceedings., pages 30–41, 2006.

[9] Alexis Bondu, Vincent Lemaire, and Barbara Poulain. Active Learning Strategies: A Case Study for Detection of Emotions in Speech. In *Advances in Data Mining. Theoretical Aspects and Applications 7th Industrial Conference, ICDM 2007, Leipzig, Germany, July 14-18, 2007. Proceedings*, pages 228–241, 2007.

[10] Sabri Boutemedjet and Djemel Ziou. A Generative Graphical Model for Collaborative Filtering of Visual Content. In *Advances in Data Mining 6th Industrial Conference on Data Mining, ICDM 2006, Leipzig, Germany, July 14-15, 2006. Proceedings*, pages 404–415, 2006.

[11] Gemma Casas-Garriga. Discovering Unbounded Episodes in Sequential Data. In *Knowledge Discovery in Databases: PKDD 2003 7th European Conference on Principles and Practice of Knowledge Discovery in Databases, Cavtat-Dubrovnik, Croatia, September 22-26, 2003, Proceedings.*, pages 83–94, 2003.

[12] Joe Catalano, Tom Armstrong, and Tim Oates. Discovering Patterns in Real-Valued Time Series. In *Knowledge Discovery in Databases: PKDD 2006 10th European Conference on Principles and Practice of Knowledge Discovery in Databases Berlin, Germany, September 18-22, 2006 Proceedings.*, pages 462–469, 2006.

[13] Michelangelo Ceci, Annalisa Appice, and Donato Malerba. Discovering emerging patterns in spatial databases: A multi-relational approach. In *Knowledge Discovery in Databases: PKDD 2007, 11th European Conference on Principles and Practice of Knowledge Discovery in Databases, Warsaw, Poland, September 17-21, 2007. Proceedings.*, pages 390–397, 2007.

[14] Saumen Chakrabarti, editor. *Mining the Web – Discovering Knowledge from Hypertext Data*. Morgan Kaufmann, 2003.

[15] Duen Horng Chau, Shashank Pandit, and Christos Faloutsos. Detecting Fraudulent Personalities in Networks of Online Auctioneers. In *Knowledge Discovery in Databases: PKDD 2006 10th European Conference on Principles and Practice of Knowledge Discovery in Databases Berlin, Germany, September 18-22, 2006 Proceedings.*, pages 103–114, 2006.

[16] Ching-Chien Chen, Snehal Thakkar, Craig Knoblock, and Cyrus Shahabi. Automatically Annotating and Integrating Spatial Datasets. In *Advances in Spatial and Temporal Databases 8th International Symposium, SSTD 2003 Santorini Island, Greece, July 24-27, 2003 Proceedings.*, pages 469–488, 2003.

[17] Zheru Chi, Hong Yan, and Tuan Pham. *Fuzzy Algorithms with Applications to Image Processing and Pattern Recognition*. World Scientific Publishing, 1996.

[18] Bill Chiu, Eamonn Keogh, and Stefano Lonardi. Probabilistic discovery of time series motifs. In *Proceedings of the 9th International Conference on Knowledge Discovery and Data Mining (SIGKDD'03)*, pages 493–498, 2003.

[19] Carlo Colombo and Alberto del Bimbo. *Image Databases: Search and Retrieval of Digital Imagery*, chapter "Visible Image Retrieval", pages 11–33. John Wiley & Sons, Inc., 2002.

[20] Nello Cristianini and John Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge University Press, 2003.

[21] Luciano da Fontoura Costa and Roberto Marcondes Cesar Jr. *Shape Analysis and Classification – Theory and Practice*. CRC Press, 2001.

[22] J. Valente de Oliveira and Witold Pedrycz. *Advances in Fuzzy Clustering and its Applications*. John Wiley and Sons, 2007.

[23] Colin DeLong, Prasanna Desikan, and Jaideep Srivastava. USER: User-Sensitive Expert Recommendations for Knowledge-Dense Environments. In *Advances in Web Mining and Web Usage Analysis: 7th International Workshop on Knowledge Discovery on the Web, WebKDD 2005, Chicago, IL, USA, August 21, 2005. Revised Papers.*, pages 77–95, 2005.

[24] Da Deng. Braving the Semantic Gap: Mapping Visual Concepts from Images and Videos. In *Advances in Data Mining – Applications in Image Mining, Medicine and Biotechnology, Management and Environmental Control, and Telecommunications - 4th Industrial Conference on Data Mining, ICDM 2004, Proceedings.*, pages 50–59, 2004.

[25] Lin Deng, Wilfred Ng, Xiaoyong Chai, and Dik-Lun Lee. Spying Out Accurate User Preferences for Search Engine Adaptation. In *Advances in Web Mining and Web Usage Analysis: 6th International Workshop on Knowledge Discovery on the Web, WebKDD 2004, Seattle, WA, USA, August 22-25, 2004, Revised Selected Papers.*, pages 87–103, 2004.

[26] Prasanna Desikan and Jaideep Srivastava. Mining Temporally Changing Web Usage Graphs. In *Advances in Web Mining and Web Usage Analysis: 6th International Workshop on Knowledge Discovery on the Web, WebKDD 2004, Seattle, WA, USA, August 22-25, 2004, Revised Selected Papers.*, pages 1–17, 2004.

[27] Inderjit Dhillon, Jacob Kogan, and Charles Nicholas. *Survey of Text Mining – Clustering, Classification and Retrieval*, chapter Feature Selection and Document Clustering, pages 73–100. Springer, 2003.

[28] Andrzej Dominik, Zbigniew Walczak, , and Jacek Wojciechowski. Classification of web documents using a graph-based model and structural patterns. In *Knowledge Discovery in Databases: PKDD 2007, 11th European Conference on Principles and Practice of Knowledge Discovery in Databases, Warsaw, Poland, September 17-21, 2007. Proceedings.*, pages 67–78, 2007.

[29] Marcelino Pereira dos Santos Silva. *Metodologia de Mineração de Padrões de Mudança em Imagens de Sensoriamento Remoto*. PhD thesis, Instituto Nacional de Pesquisas Espaciais, 2006.

[30] Hicham G. Elmongui, Mohamed F. Mokbel, and Walid G. Aref. Spatio-temporal Histograms. In *Advances in Spatial and Temporal Databases 9th International Symposium, SSTD 2005, Angra dos Reis, Brazil, August 22-24, 2005. Proceedings.*, pages 19–36, 2005.

[31] Laurene V. Fausett. *Fundamentals of Neural Networks*. Prentice Hall, 1994.

[32] Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasamy Uthurusamy, editors. *Advances in Knowledge Discovery and Data Mining*. MIT Press, 1st edition, 1996.

[33] Ronen Feldman. *The Handbook of Data Mining*, chapter Mining Text Data, pages 52–95. Lawrence Erlbaum Associates Publishers, 2003.

[34] Ronen Feldman and James Sanger, editors. *The Text Mining Handbook – Advanced Approaches in Analyzing Unstructured Data*. Cambridge University Press, 2006.

[35] Ling Feng, Rogier Brussee, Henk Blanken, and Mettina Veenstra. *Multimedia Retrieval*, chapter Languages for Metadata, pages 23–51. Springer, 2007.

[36] Richard Frank, Wen Jin, and Martin Ester. Efficiently Mining Regional Outliers in Spatial Data. In *Advances in Spatial and Temporal Databases 10th International Symposium, SSTD 2007, Boston, MA, USA, July 16-18, 2007. Proceedings.*, pages 112–129, 2007.

[37] Elias Frentzos. Indexing Objects Moving on Fixed Networks. In *Advances in Spatial and Temporal Databases 8th International Symposium, SSTD 2003 Santorini Island, Greece, July 24-27, 2003 Proceedings.*, pages 289–305, 2003.

[38] Elias Frentzos, Kostas Gratsias, Nikos Pelekis, , and Yannis Theodoridis. Nearest Neighbor Search on Moving Object Trajectories. In *Advances in Spatial and Temporal Databases 9th International Symposium, SSTD 2005, Angra dos Reis, Brazil, August 22-24, 2005. Proceedings.*, pages 328–345, 2005.

[39] James Geller, Sapankumar Parikh, and Sriram Krishnan. Blog mining for the Fortune 500. In *Machine Learning and Data Mining in Pattern Recognition: 5th International Conference, MLDM 2007, Leipzig, Germany, July 18-20, 2007. Proceedings.*, pages 379–391, 2007.

[40] André Ricardo Abed Gregio, Rafael Santos, and Antonio Montes Filho. Evaluation of data mining techniques for suspicious network activity classification using honeypots data. In *Proceedings of SPIE Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security, 2007, Orlando, Florida, EUA.*, 2007.

[41] Dihua Guo, Hui Xiong, Vijay Atluri, and Nabil Adam. Semantic Feature Selection for Object Discovery in High-Resolution Remote Sensing Imagery. In *Advances in Knowledge Discovery and Data Mining 11th Pacific-Asia Conference, PAKDD 2007, Nanjing, China, May 22-25, 2007. Proceedings.*, pages 71–83, 2007.

[42] Jaana Heino and Hannu Toivonen. Automated Detection of Epidemics from the Usage Logs of a Physicians' Reference Database. In *Knowledge Discovery in Databases: PKDD 2003 7th European Conference on Principles and Practice of Knowledge Discovery in Databases, Cavtat-Dubrovnik, Croatia, September 22-26, 2003, Proceedings.*, pages 180–191, 2003.

[43] Stefan Holland, Martin Ester, and Werner Kießling. Preference Mining: A Novel Approach on Mining User Preferences for Personalized Applications. In *Knowledge Discovery in Databases: PKDD 2003 7th European Conference on Principles and Practice of Knowledge Discovery in Databases, Cavtat-Dubrovnik, Croatia, September 22-26, 2003, Proceedings.*, pages 204–216, 2003.

[44] Te-Ming Huang, Vojislav Kecman, and Ivica Kopriva. *Kernel Based Algorithms for Mining Huge Data Sets*. Springer, 2006.

[45] Yu Huang, Joan Llach, and Sitaram Bhagavathy. Players and Ball Detection in Soccer Videos Based on Color Segmentation and Shape Analysis. In *Multimedia Content Analysis and Mining International Workshop, MCAM 2007, Weihai, China, June 30-July 1, 2007. Proceedings.*, pages 416–425, 2007.

[46] Bernard Hugueney. Adaptive Segmentation-Based Symbolic Representations of Time Series for Better Modeling and Lower Bounding Distance Measures. In *Knowledge Discovery in Databases: PKDD 2006 10th European Conference on Principles and Practice of Knowledge Discovery in Databases Berlin, Germany, September 18-22, 2006 Proceedings.*, pages 545–552, 2006.

[47] Shin ichi Kobayashi, Yasuyuki Shirai, Kazuo Hiyane, Fumihiro Kumeno, Hiroshi Inujima, and Noriyoshi Yamauchi. Technology Trends Analysis from the Internet Resources. In *Advances in Knowledge Discovery and Data Mining 9th Pacific-Asia Conference, PAKDD 2005, Hanoi, Vietnam, May 18-20, 2005. Proceedings.*, pages 820–825, 2005.

[48] Peter Jackson. *Introduction to Expert Systems*. Addison-Wesley, 1986.

[49] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Comput. Surv.*, 31(3):264–323, 1999.

[50] Buhwan Jeong, Daewon Lee, and Hyunbo Choand Boonserm Kulvatunyou. A Kernel Method for Measuring Structural Similarity Between XML Documents. In *New Trends in Applied Artificial Intelligence 20th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, IEA-AIE 2007, Kyoto, Japan, June 26-29, 2007. Proceedings.*, pages 572–581, 2007.

[51] Tao Jiang and Ah-Hwee Tan. Discovering Image-Text Associations for Cross-Media Web Information Fusion. In *Knowledge Discovery in Databases: PKDD 2006 10th European Conference on Principles and Practice of Knowledge Discovery in Databases Berlin, Germany, September 18-22, 2006 Proceedings.*, pages 561–568, 2006.

[52] Panos Kalnis, Nikos Mamoulis, and Spiridon Bakiras. On Discovering Moving Clusters in Spatio-temporal Data. In *Advances in Spatial and Temporal Databases 9th International Symposium, SSTD 2005, Angra dos Reis, Brazil, August 22-24, 2005. Proceedings.*, pages 364–381, 2005.

[53] Mehmed Kantardzic. *Data Mining: Concepts, Models, Methods, and Algorithms*. John Wiley & Sons, 2003.

[54] Jun Karamon, Yutaka Matsuo, Hikaru Yamamoto, and Mitsuru Ishizuka. Generating social network features for link-based classification. In *Knowledge Discovery in Databases: PKDD 2007, 11th European Conference on Principles and Practice of Knowledge Discovery in Databases, Warsaw, Poland, September 17-21, 2007. Proceedings.*, pages 127–139, 2007.

[55] Steffen Kempe and Jochen Hipp. Mining Sequences of Temporal Intervals. In *Knowledge Discovery in Databases: PKDD 2006 10th European Conference on Principles and Practice of Knowledge Discovery in Databases Berlin, Germany, September 18-22, 2006 Proceedings.*, pages 569–576, 2006.

[56] Eamonn Keogh. Efficiently Finding Arbitrarily Scaled Patterns in Massive Time Series Databases. In *Knowledge Discovery in Databases: PKDD 2003 7th European Conference on Principles and Practice of Knowledge Discovery in Databases, Cavtat-Dubrovnik, Croatia, September 22-26, 2003, Proceedings.*, pages 253–265, 2003.

[57] Eamonn J. Keogh and Michael J. Pazzani. Scaling up Dynamic Time Warping to Massive Datasets. In *Principles of Data Mining and Knowledge Discovery, Third European Conference, PKDD99, Prague, Czech Republic, September 15-18, 1999. Proceedings.*, pages 1–11, 1999.

[58] Teuvo Kohonen. *Self-Organizing Maps*. Springer, 2nd edition, 1997.

[59] Aleksander Kolcz and Wen tau Yih. Site-independent template-block detection. In *Knowledge Discovery in Databases: PKDD 2007, 11th European Conference on Principles and Practice of Knowledge Discovery in Databases, Warsaw, Poland, September 17-21, 2007. Proceedings.*, pages 152–163, 2007.

[60] Krzysztof Koperski and Jiawei Han. Discovery of spatial association rules in geographic information databases. In *Advances in spatial databases: 4th international symposium, SSD '95, Portland, ME, USA, August 6–9, 1995: proceedings*, pages 47–66, 1995.

[61] Michal Kozielski. Multilevel conditional fuzzy c-means clustering of xml documents. In *Knowledge Discovery in Databases: PKDD 2007, 11th European Conference on Principles and Practice of Knowledge Discovery in Databases, Warsaw, Poland, September 17-21, 2007. Proceedings.*, pages 532–539, 2007.

[62] Daniel T. Larose. *Discovering Knowledge in Data – An Introduction to Data Mining*. Wiley-Interscience, 2005.

[63] Mark Last, Abraham Kandel, and Horst Bunke, editors. *Data Mining in Time Series Databases*. World Scientific, 2004.

[64] Julien Law-To, Valérie Gouet-Brunet, Olivier Buisson, and Nozha Boujemaa. Labeling Complementary Local Descriptors Behavior for Video Copy Detection. In *Multimedia Content Representation, Classification and Security International Workshop, MRCS 2006, Istanbul, Turkey, September 11-13, 2006. Proceedings*, pages 290–297, 2006.

[65] Haojie Li, Si Wu, Shan Ba, Shouxun Lin, and Yongdong Zhang. Automatic Detection and Recognition of Athlete Actions in Diving Video. In *Advances in Multimedia Modeling 13th International Multimedia Modeling Conference, MMM 2007, Singapore, January 9-12, 2007. Proceedings, Part II*, pages 73–82, 2007.

[66] Xiaolei Li, Jiawei Han, Jae-Gil Lee, and Hector Gonzalez. Traffic Density-Based Discovery of Hot Routes in Road Networks. In *Advances in Spatial and Temporal Databases 10th International Symposium, SSTD 2007, Boston, MA, USA, July 16-18, 2007. Proceedings.*, pages 441–459, 2007.

[67] Jessica Lin, Eamonn Keogh, Stefano Lonardi, and Pranav Patel. Finding motifs in time series. In *Proceedings of the Second Workshop on Temporal Data Mining, at the 8th ACM SIGKDD*, pages 53–68, 2002.

[68] Lu Liu, Wei Lai, Xian-Sheng Hua, and Shi-Qiang Yang. Video Histogram: A Novel Video Signature for Efficient Web Video Duplicate Detection. In *Advances in Multimedia Modeling 13th International Multimedia Modeling Conference, MMM 2007, Singapore, January 9-12, 2007. Proceedings, Part II*, pages 94–103, 2007.

[69] Ying Liu, Dengsheng Zhang, and Guojun Lu. SIEVE – Search Images Effectively Through Visual Elimination. In *Multimedia Content Analysis and Mining International Workshop, MCAM 2007, Weihai, China, June 30-July 1, 2007. Proceedings.*, pages 381–390, 2007.

[70] Carl G. Looney. *Pattern Recognition Using Neural Networks*. Oxford University Press, 1st edition, 1997.

[71] Hua Lu, Zhiyong Huang, Christian S. Jensen, and Linhao Xu. Distributed, Concurrent Range Monitoring of Spatial-Network Constrained Mobile Objects. In *Advances in Spatial and Temporal Databases 10th International Symposium, SSTD 2007, Boston, MA, USA, July 16-18, 2007. Proceedings.*, pages 403–422, 2007.

[72] Lin Lu, Margaret Dunham, and Yu Meng. Mining Significant Usage Patterns from Clickstream Data. In *Advances in Web Mining and Web Usage Analysis: 7th International Workshop on Knowledge Discovery on the Web, WebKDD 2005, Chicago, IL, USA, August 21, 2005. Revised Papers.*, pages 1–17, 2005.

[73] Donato Malerba, Michelangelo Ceci, and Annalisa Appice. SVM-Based Audio Classification for Content-Based Multimedia Retrieval. In *Knowledge Discovery*

in Databases: PKDD 2005 9th European Conference on Principles and Practice of Knowledge Discovery in Databases, Porto, Portugal, October 3-7, 2005. Proceedings., pages 169–180, 2005.

[74] Shotaro Matsumoto, Hiroya Takamura, and Manabu Okumura. Sentiment Classification Using Word Sub-sequences and Dependency Sub-trees. In *Advances in Knowledge Discovery and Data Mining 9th Pacific-Asia Conference, PAKDD 2005, Hanoi, Vietnam, May 18-20, 2005. Proceedings.*, pages 301–311, 2005.

[75] Alessandro Micarelli and Giuseppe Sansonetti. A case-based approach to anomaly intrusion detection. In *Machine Learning and Data Mining in Pattern Recognition: 5th International Conference, MLDM 2007, Leipzig, Germany, July 18-20, 2007. Proceedings.*, pages 434–448, 2007.

[76] Mikołaj Morzy. Mining frequent trajectories of moving objects for location prediction. In *Machine Learning and Data Mining in Pattern Recognition: 5th International Conference, MLDM 2007, Leipzig, Germany, July 18-20, 2007. Proceedings.*, pages 667–680, 2007.

[77] Jinfeng Ni and Chinya V. Ravishankar. PA-Tree: A Parametric Indexing Scheme for Spatio-temporal Trajectories. In *Advances in Spatial and Temporal Databases 9th International Symposium, SSTD 2005, Angra dos Reis, Brazil, August 22-24, 2005. Proceedings.*, pages 254–272, 2005.

[78] Mordechai Nisenson, Ido Yariv, Ran El-Yaniv, , and Ron Meir. Towards Behaviometric Security Systems: Learning to Identify a Typist. In *Knowledge Discovery in Databases: PKDD 2003 7th European Conference on Principles and Practice of Knowledge Discovery in Databases, Cavtat-Dubrovnik, Croatia, September 22-26, 2003, Proceedings.*, pages 363–374, 2003.

[79] Mark Nixon and Alberto Aguado. *Feature Extraction and Image Processing*. Newnes, 2002.

[80] Nicola Orio. "music retrieval: A tutorial and review". *Foundations and Trends in Information Retrieval*, 1(1):1–90, 2006.

[81] H.T. Pao, Y.Y. Xu, S.C. Chung, and H.C. Fu. Constructing and application of multimedia TV news archives. In *Multimedia Content Analysis and Mining International Workshop, MCAM 2007, Weihai, China, June 30-July 1, 2007. Proceedings.*, pages 151–160, 2007.

[82] Witold Pedrycz. *Knowledge-Based Clustering – From Data to Information Granules*. Wiley-Interscience, 2005.

[83] Petra Perner. *Data Mining on Multimedia Data*, volume 2558 of *Lecture Notes in Computer Science*. Springer-Verlag Inc., New York, NY, USA, 2002.

[84] Petra Perner, Horst Perner, Angela Bühring, and Silke Jänichen. Mining Images to Find General Forms of Biological Objects. In *Advances in Data Mining – Applications in Image Mining, Medicine and Biotechnology, Management and Environmental Control, and Telecommunications - 4th Industrial Conference on Data Mining, ICDM 2004, Proceedings.*, pages 60–68, 2004.

[85] Octavian Procopiuc, Pankaj K. Agarwal, Lars Arge, and Jeffrey Scott Vitter. Bkd-Tree: A Dynamic Scalable kd-Tree. In *Advances in Spatial and Temporal Databases 8th International Symposium, SSTD 2003 Santorini Island, Greece, July 24-27, 2003 Proceedings.*, pages 46–65, 2003.

[86] Dorian Pyle. *Data Preparation for Data Mining*. Academic Press, 1st edition, 1999.

[87] J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[88] Elena Ranguelova and Mark Huiskes. *Multimedia Retrieval*, chapter Pattern Recognition for Multimedia Content Analysis, pages 52–95. Springer, 2007.

[89] John A. Richards. *Remote Sensing Digital Image Analysis – An Introduction*. Springer-Verlag, 1993.

[90] Mirek Riedewald, Divyakant Agrawal, Amr El Abbadi, and Flip Korn. Accessing Scientific Data: Simpler is Better. In *Advances in Spatial and Temporal Databases 8th International Symposium, SSTD 2003 Santorini Island, Greece, July 24-27, 2003 Proceedings.*, pages 214–232, 2003.

[91] Yong Rui and Guo-Jun Qi. Learning concepts by modeling relationships. In *Multimedia Content Analysis and Mining – International Workshop, MCAM 2007*, pages 5–13, 2007.

[92] Reza Sadoddin and Ali A. Ghorbani. A comparative study of unsupervised machine learning and data mining techniques for intrusion detection. In *Machine Learning and Data Mining in Pattern Recognition: 5th International Conference, MLDM 2007, Leipzig, Germany, July 18-20, 2007. Proceedings.*, pages 404–418, 2007.

[93] M. De Santo, G. Percannella, C. Sansone, and M. Vento. Unsupervised News Video Segmentation by Combined Audio-Video Analysis. In *Multimedia Content Representation, Classification and Security International Workshop, MRCS 2006, Istanbul, Turkey, September 11-13, 2006. Proceedings*, pages 273–281, 2006.

[94] Rafael Santos. Weka na Munheca – Um guia para uso do Weka em scripts e integração com aplicações em Java. http://www.lac.inpe.br/∼rafael.santos/Docs/CAP359/2005/weka.pdf, 2005. Visitado em Setembro de 2008.

[95] Rafael Santos. *Computação e Matemática Aplicada às Ciências e Tecnologias Espaciais*, chapter Introdução à Mineração de Dados com Aplicações em Ciências

Ambientais e Espaciais, pages 15–38. Instituto Nacional de Pesquisas Espaciais, 2008.

[96] Rafael Santos. Material da disciplina CAP-359: Princípios e aplicações de mineração de dados. http://www.lac.inpe.br/∼rafael.santos/cap359.jsp, 2008. Visitado em Setembro de 2008.

[97] Rafael Santos, Takeshi Ohashi, Takaichi Yoshida, and Toshiaki Ejima. Biased clustering method for partially supervised classification. In *Proceedings of SPIE Nonlinear Image Processing IX*, pages 174–185, 1998.

[98] Mika Sato-Ilic and Lakhmi C. Jain. *Innovations in Fuzzy Clustering*. Springer, 2006.

[99] Philipp Schügerl, Robert Sorschag, Werner Bailer, and Georg Thallinger. Object re-detection using SIFT and MPEG-7 color descriptors. In *Multimedia Content Analysis and Mining International Workshop, MCAM 2007, Weihai, China, June 30-July 1, 2007. Proceedings.*, pages 305–314, 2007.

[100] Nicu Sebe, Yuncai Liu, Yueting Zhuang, and Thomas S. Huang, editors. *Multimedia Content Analysis and Mining – International Workshop, MCAM 2007*, volume 4577 of *Lecture Notes in Computer Science*, New York, NY, USA, 2007. Springer-Verlag Inc.

[101] Christian Simmermacher, Da Deng, and Stephen Cranefield. Feature Analysis and Classification of Classical Musical Instruments: An Empirical Study. In *Advances in Data Mining 6th Industrial Conference on Data Mining, ICDM 2006, Leipzig, Germany, July 14-15, 2006. Proceedings*, pages 444–458, 2006.

[102] Bhushan Shankar Suryavanshi, Nematollaah Shiri, and Sudhir P. Mudur. Adaptive Web Usage Profiling. In *Advances in Web Mining and Web Usage Analysis: 7th International Workshop on Knowledge Discovery on the Web, WebKDD 2005, Chicago, IL, USA, August 21, 2005. Revised Papers.*, pages 119–138, 2005.

[103] Xiaofeng Tong, Tao Wang, Wenlong Li, Yimin Zhang, Bo Yang, Fei Wang, Lifeng Sun, and Shiqiang Yang. A three-level scheme for real-time ball tracking. In *Multimedia Content Analysis and Mining International Workshop, MCAM 2007, Weihai, China, June 30-July 1, 2007. Proceedings.*, pages 161–171, 2007.

[104] B. Tso and P. M. Mather. *Classification Methods for Remotely Sensed Data*. Taylor and Francis, London, 2000.

[105] Shusaku Tsumoto. Rule Discovery in Large Time-Series Medical Databases. In *Principles of Data Mining and Knowledge Discovery, Third European Conference, PKDD99, Prague, Czech Republic, September 15-18, 1999. Proceedings.*, pages 23–31, 1999.

[106] Subodh Vaid, Christopher B. Jones, Hideo Joho, and Mark Sanderson. Spatio-textual Indexing for Geographical Search on the Web. In *Advances in Spatial and*

Temporal Databases 9th International Symposium, SSTD 2005, Angra dos Reis, Brazil, August 22-24, 2005. Proceedings., pages 218–235, 2005.

[107] Sudha Velusamy, Balaji Thoshkahna, and K.R. Ramakrishnan. A Novel Melody Line Identification Algorithm for Polyphonic MIDI Music. In *Advances in Multimedia Modeling 13th International Multimedia Modeling Conference, MMM 2007, Singapore, January 9-12, 2007. Proceedings, Part II*, pages 248–257, 2007.

[108] Anne-Marie Vercoustre, Mounir Fegas, Saba Gul, and Yves Lechevallier. A Flexible Structured-Based Representation for XML Document Mining. In *Advances in XML Information Retrieval and Evaluation - 4th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2005. Proceedings.*, pages 443–457, 2006.

[109] Chaokun Wang, Jianmin Wang, Jianzhong Li, Jia-Guang Sun, and Shengfei Shi. MuSQL: A Music Structured Query Language. In *Advances in Multimedia Modeling 13th International Multimedia Modeling Conference, MMM 2007, Singapore, January 9-12, 2007. Proceedings, Part II*, pages 216–225, 2007.

[110] Cuiru Wang, Hejin Yuan, Jun Liu, Tao Zhou, and Huiling Lu. A Novel Support Vector Machine Ensemble Based on Subtractive Clustering Analysis. In *Advances in Knowledge Discovery and Data Mining 11th Pacific-Asia Conference, PAKDD 2007, Nanjing, China, May 22-25, 2007. Proceedings.*, pages 849–856, 2007.

[111] Jinqiao Wang, Qingshan Liu, Lingyu Duan, Hanqing Lu, and Changsheng Xu. Automatic TV Logo Detection, Tracking and Removal in Broadcast Video. In *Advances in Multimedia Modeling 13th International Multimedia Modeling Conference, MMM 2007, Singapore, January 9-12, 2007. Proceedings, Part II*, pages 63–72, 2007.

[112] Julie Wilson. Automated Classification of Images from Crystallisation Experiments. In *Advances in Data Mining 6th Industrial Conference on Data Mining, ICDM 2006, Leipzig, Germany, July 14-15, 2006. Proceedings*, pages 459–473, 2006.

[113] René Witte. Introduction to Text Mining. http://www.edbt2006.de/edbt-share/IntroductionToTextMining.pdf, 2006.

[114] Ian H. Witten and Eibe Frank. *Data Mining - Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann Publishers, 2000.

[115] Limin Xu, Zhenmin Tang, Keke He, and Bo Qian. Transformation-Based GMM with Improved Cluster Algorithm for Speaker Identification. In *Advances in Knowledge Discovery and Data Mining 11th Pacific-Asia Conference, PAKDD 2007, Nanjing, China, May 22-25, 2007. Proceedings.*, pages 1006–1014, 2007.

[116] Albert K.W. Yeung and G. Brent Hall. *Spatial Database Systems – Design, Implementation and Project Management*. Springer, 2007.

[117] Shengyang Yu, Yan Zhang, Yonggang Wang, and Jie Yang. Color-texture image segmentation by combining region and photometric invariant edge information. In *Multimedia Content Analysis and Mining International Workshop, MCAM 2007, Weihai, China, June 30-July 1, 2007. Proceedings.*, pages 286–294, 2007.

[118] Osmar R. Zaïane, Simeon J. Simoff, and Chabane Djeraba, editors. *Mining Multimedia and Complex Data – KDD Workshop MDM/KDD 2002, PAKDD Workshop KDMCD 2002*, volume 2797 of *Lecture Notes in Computer Science*, New York, NY, USA, 2003. Springer-Verlag Inc.

[119] Yi Zhang and Bing Liu. Semantic text classification of emergent disease reports. In *Knowledge Discovery in Databases: PKDD 2007, 11th European Conference on Principles and Practice of Knowledge Discovery in Databases, Warsaw, Poland, September 17-21, 2007. Proceedings.*, pages 629–637, 2007.

[120] Yaqin Zhao, Xianzhong Zhou, and Guizhong Tang. Audiovisual Integration for Racquet Sports Video Retrieval. In *Advanced Data Mining and Applications Second International Conference, ADMA 2006, Proceedings.*, pages 673–680, 2006.

[121] Yingying Zhu, Zhong Ming, and Qiang Huang. SVM-Based Audio Classification for Content-Based Multimedia Retrieval. In *Multimedia Content Analysis and Mining International Workshop, MCAM 2007, Weihai, China, June 30-July 1, 2007. Proceedings.*, pages 474–482, 2007.